A+ A NICE MIX OF THEORY AND PRACTICE.

LSC

A REPORT ON ENVIRONMENT MODELING AND MODEL PRE-PROCESSING FOR JASON, THE BERKELEY ROBOT

by

Diana Templeton Killen
and
Peter John Blatman

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
C.S. 292H
Winter 1975
Prof. L.S. Coles

Abstract

This paper is a report in two parts. The first proposes extensions to Jason's relational model in order to enable Jason to cope with a larger class of real world environments.

The second part deals with the application of cluster analysis to grid map pre-processing. This work was undertaken in an effort to incorporate groups of closely spaced objects into single "pseudo-objects", thereby simplifying subsequent navigation and path computation. In addition, object clustering has the potential to reduce the number of turns needed in calculated robot trajectories, these turns being the major source of accumulated error.

While not implemented in Fortran, all clustering routines were written and fully debugged in APL. (The programs were written however, with an eye towards subsequent conversion to Fortran). Test runs were made on actual Cory Hall room data, which was painstakingly collected.

TABLE OF CONTENTS

PART I. ENVIRONMENT MODELING AND DATA COLLECTION

INTRODUCTION

Detailed measurements of the third floor of Cory Hall were collected so that JASON, the Berkeley robot, would have accurate information on the environment of the third floor . It soon became obvious that the current relational and grid models for JASON were inadequate for accurately modeling Cory Hall . Hence, the first part of this section deals with the modifications and extensions to the relational model that are needed to more accurately model the real environment . The second part of this section deals with the actual collection of data , including information collected which was used as input to the cluster analysis program described in detail in part II of this report .

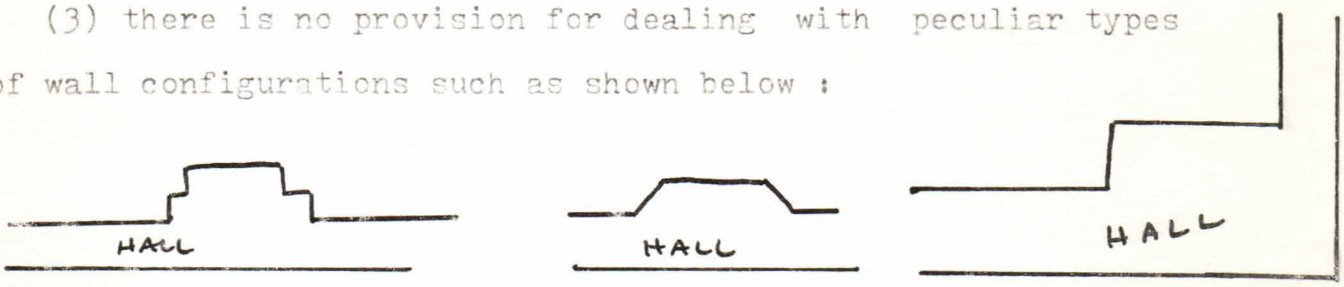
EXTENSIONS TO THE RELATIONAL MODEL

Jason's current relational model cannot handle certain information about the third floor of Cory Hall . In particular, the primary troublespots are that

(1) some corners do not form right angles

(2) there is no provision in the model for the openings that occur where two or more hallways intersect and

(3) there is no provision for dealing with peculiar types of wall configurations such as shown below :



One solution , of course , is to use a completely different sort of model, such as a line segment model rather than the grid and relational models that are currently used. However, I have assumed that it is desirable to continue using the relational and grid models and thus, I will outline the necessary changes and extensions that are necessary to enable a realistic representation of the third floor of Cory Hall.

First of all, to solve problem (2) above, hall nodes in the relational model should have an additional field called HL OPEN (hall opening) which would point to a list of nodes of type OPENING, which are analagous to the current field, DOOR PTR, which points to a list of doors .

Each node of type OPENING would then have the following fields :

OPN NAME - name of the opening

OPN WIDTH - width of the opening

OPN HALL1 - pointer to the node of area 1 that the opening connects

OPN HALL2 - pointer to the node of area 2 that the opening connects

OPN1 LOC X - x-coordinate of the center of the opening in area 1

OPN1 LOC Y - y-coordinate of the center of the opening in area 1

OPN2 LOC X - x-coordinate of the center of the opening in area 2

OPN2 LCc Y - y-coordinate of the center of the opening in area 2

REL ANGL - relative angle of area 2's x-axis based on the x-axis of area 1

OPN NEXT - pointer to the next node of the same type

Next, I propose another node type for walls. A wall in this context refers to a stationary object that fills up a room or hall along its perimeter, where the perimeter of the room or hall is what has been previously called "the maximum enclosing rectangle". Hence, rooms or halls are now represented as rectangular, where the "maximum enclosing rectangle " is infact, the rectangular room or hall. A room or hall then contains two types of objects: (1) normal objects as defined in the current model and (2) objects of type WALL, which essentially fill the room to give the room or hall its apparent contours. Note that if desired, it is possible to utilize the new node type ofwall to handle any immovable object whether or not the "object" is on the perimeter of the room. (such as fixed podiums) In addition, so that corners that do not form right angles can be represented, it is desirable to allow objects of type wall to be non-rectangular. Thus each wall node will have a field that points to a list of vertices that represent the "wall". Each vertex corresponds to a (x,y) pair. It should also be pointed out that a separate node type for recesses is no longer necessary if nodes of type wall are used. Also, the fields RM X LEN, RM Y LEN, and RCES PTR may be deleted from room and hall nodes. A new field called WALL PTR should be added to room and hall nodes. WALL PTR would point to list of type wall which are the "walls" of the room.

The wall nodes would have the following fields :

WALL NAME - name of the wall

WALL AREA - pointer to the node of the area (room or hall) which contains this wall

VERT PTR - pointer to a list of vertices for this wall

WALL NEXT - pointer to the next wall in this room or hall

DATA COLLECTION

Two types of measurements were collected on Cory Hall ; the first type consisted of measurements of various classrooms and their contents for use by the clustering program discussed in the second part of this report. The second set of measurements consists of additional measurements of the third floor of Cory Hall which are necessary for a relatively accurate description of the "real" environment.

Three classrooms were measured extensively in order to present realistic data to the clustering program. Since there is only one classroom on the third floor of Cory Hall (room 395), two additional classrooms on the second floor were measured. (rooms 293 and 237). For each room, the measurements collected included placement of doors, walls, tables, chairs, desks, ~~garbage~~ wastepaper cans, podiums and any other objects such as pipes that appeared in the room. The measurements were transformed to coordinates with the south-western corner of the room chosen as (0,0). Each object was represented by a list of its vertices. See Appendix I for more details.

The additional measurements that were made but not used by the clustering program consist of detailed measurements of the hallways of the third floor of Cory Hall. Besides the measurements of the contours and orientation of the hallways, information was collected on all objects both movable and immovable that appeared in the relevant halls. The raw data appears in Appendix IV.

CONCLUSION

Some future work must be done before the data can be used by Jason in building his environment model. In particular, the relational model must be modified so that the changes and extensions proposed in the preceding pages are actually operational. Also, the raw data presented in Appendix IV will have to be transformed slightly (calculation of coordinates and such) to a form that Jason can handle.

PART II. GRID MAP PRE-PROCESSING

I. Introduction

The purpose of this project was to experiment with the possibility (and practicability) of improving Jason's overall navigational performance and accuracy through grid-map pre-processing. In the performance of successive GO TO LOC type directives, error in the robot's conception of its relative location accumulates in the form of wheel count inaccuracy. This type of hardware generated error tends to place limitations on Jason's usefulness in applications requiring continuous operation. explain

A major source of this type of error is inherent in the performance of turns, due to the fact that turning requires a series of non-orthogonal movements, with the rear wheel clutch released. Since turns need only be performed when obstacles in a computed path are encountered, (aside from a possible initial turn), it would be desirable to minimize the number of obstacles which are "apparent" to the robot. In order to accomplish this, existing obstacles in the environment (and apparent to the robot through the grid-map ~~which is built up by the~~ relational model) must be somehow "clustered" together to form "pseudo-objects."

This paper describes the details of a pseudo-object generation system, which has been implemented in APL. The system uses the graph-theoretic concept of a "minimal spanning tree" to perform the object clustering, and the concept of the "convex hull" of a set of planar points to generate the pseudo-object corresponding to a group of objects.

II. Object Clustering

In order to describe the method used to cluster object groups into pseudo-objects, a few concepts must be introduced.

Def. - A tree T is said to be a spanning tree of a connected graph G if T is a subgraph of G and T contains all vertices of G. (This may be thought of as a set of edges in a connected graph, whereby any point can be reached from any other point).

Def. - If a connected graph G is weighted (a real number is associated with each edge in G), then the weight of a spanning tree T of G, is the sum of the weights of the edges of T.

Def. - A spanning tree T of a weighted connected graph G is a minimal spanning tree of G if and only if there exists no other spanning tree of G whose weight is less than the weight of T.

The procedure which is used to cluster objects into groups treats each object as a node in a completely connected graph. For simplicity in computation, each object may be viewed as consisting of only a single point, which is the mean of all its points. The weight of an edge joining two points is defined as the euclidean distance between the two points. (This may lead to some problems, which will be discussed in the conclusion of this paper.) A minimal spanning tree is then generated, which connects all object midpoints together. Each edge in the spanning tree is then compared with some threshold criterion (typically average edge weight), and edges which surpass this threshold are deemed "inconsistent" and are deleted from the tree. By deleting n edges in this manner, n+1 strongly connected components of the spanning tree will be formed. The set of objects associated with the points which comprise each of these components forms a cluster set. (i.e., each strongly connected component is a

↳ define?

cluster set. ~~(see fig. 1)~~ The alg was taken from Deo^3

Minimal Spanning Tree Algorithm (from Deo [3])

1. Compute the midpoints of the n objects and label these V1, V2,..., Vn
2. Compute Dij, the euclidean distance between Vi and Vj $(i \neq j)$ for all i and $j \leq n$.
3. Initialize subtree S as vertex V1.
4. Connect S to its nearest neighboring vertex (where all distances between vertices in S and all other unconnected vertices are considered) and define this new subtree as S.
5. If all n vertices have been connected by n-1 edges then halt, these edges form a minimal spanning tree. Otherwise, go to step 4.

Once the minimal spanning tree has been generated, inconsistent edges must be detected and deleted. Zahn [5] considers this problem for a variety of clustering applications, but experimentally, I found that one standard deviation from the average edge weight formed a good threshold.

Having deleted inconsistent edges, the strongly connected vertices remaining (which form the clusters) must be separated from one another. Warshall's well known transitive closure algorithm [4] was used to construct the "reachability matrix" for the set of object midpoints. A simple recursive search procedure was then implemented to scan this matrix for the points belonging to each cluster.

Assuming that Warshall's algorithm for transitive closure has generated the matrix M, where Mij=1 if and only if Vi is strongly connected to Vj, the following algorithm will separate the vertices into cluster sets:

1. Construct a list L, where Li= 1 if and only if Vi has not been put into a cluster set, and 0 otherwise. Initialize L to a list of n ONES ~~zeroes,~~ where M is an n x n matrix.
2. Set i=1.
3. For all j, such that Mij=1 (denote these indices J1, J2, ...Jk), set Lj=0, and Li=0.
4. Vi, VJ1, VJ2,...,VJk form a cluster.
5. If all entries in L are marked zero, halt, all clusters have been formed. Otherwise set i= the index of the first 1 entry in L, and return to step 3.

SPANNING TREE LINKS
237 CORY

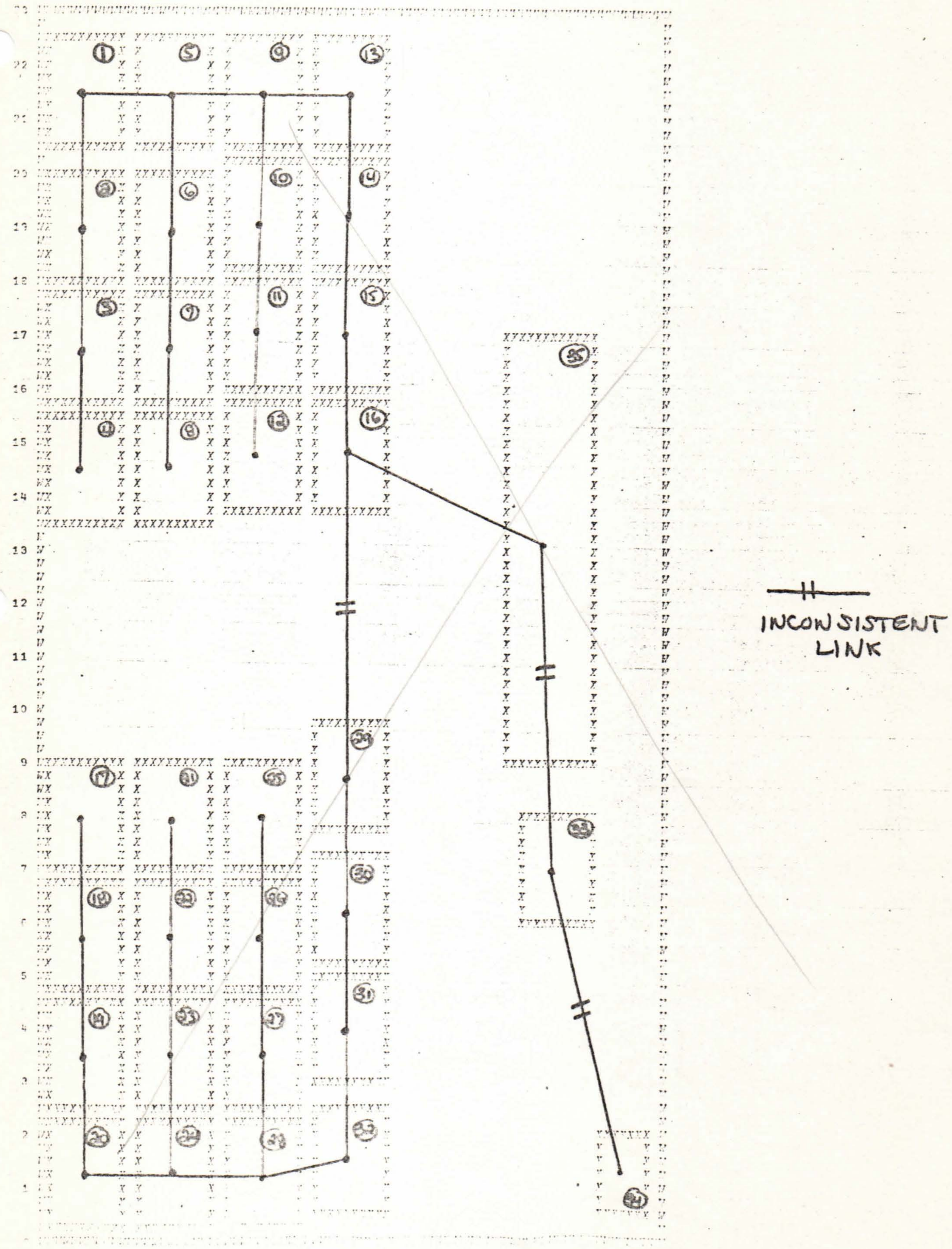


Fig. 1 - Spanning Tree Generation

III. Pseudo-object generation

Once the objects in a cluster set have been determined, it remains only to generate a pseudo-object based on the objects in the cluster set. This in itself is a very difficult problem. Ideally, the pseudo-object should not occupy a great deal more space than the objects which it encompasses. An edge-tracing type approach will give good space reduction characteristics, but can not guarantee inclusion of all objects. A convex-hull approach will guarantee inclusion of all ↳why? objects in a cluster, but may have the unfortunate side effect of large reductions in free space. ✓

I decided to go the convex-hull route~~,~~ and experiment with various hull-segment generation algorithms~~,~~ in an attempt to overcome the space reduction problem. I used the convex hull algorithm given in Graham[8] (which I found ~~had a~~ TO HAVE serious error, and corrected it).

<u>Algorithm for determining the convex hull of a set of points in two dimensions</u>

1. Compute a point X0 which must lie inside the hull (take avg. of all pt
2. Translate all points rectilinearly so that x0 lies at (0,0).
3. Convert all points to polar coordinates, where the angle of each point is measured from the half line emanating from x0 in the positive direction.
4. Order all points with respect to increase in angle. If two or more points have equal angular displacements, retain only that point with the greatest magnitude. Call this set Of points S.
5. Start with three consecutive points in S, Pi, Pj, and Pk. where $\theta 1<\theta 2<\theta 3$. (see fig.2)

6. (Refer to fig. 3) With respect to Pj, two cases exist:

```
(i) $\alpha+\beta>\pi$ Clearly Pj cannot lie on the convex hull.
    (ia) Mark Pj as deleted from S.
    (ib) Replace Pj with Pi.
    (ic) Replace Pi with P(i-1) (where P0=Pn)
    (id) If Pi has been marked deleted, go back to (ic) -(Graham omitted this)
    (ie) Go back to beginning of 6.

(ii) $\alpha+\beta\leq\pi$ Pj lies potentially on the convex hull.
    (iia) Mark Pj as having been considered.
    (iib) Replace Pi with Pj
    (iic) Replace Pj with Pk
    (iid) Replace Pk with P(k+1) (where P(n+1)=P1)
    (iie) If all points have been marked either deleted or
          considered, then halt, all points marked considered
          are on the convex hull. Otherwise, go back to
          beginning of 6.
```

In order to generate a pseudo-object for the grid map, the extremal points of the objects in a cluster set (obviously no other points need be considered) are taken as input to the convex hull algorithm. The points on the hull are then joined pairwise in order by line segments, thus creating an enclosing polygon for the objects in the cluster set. This polygon is considered to be the pseudo-object.

Since the grid map for Jason is discretized, a point on a line may be represented only by some ordered pair $(i\Delta, j\Delta)$, where i and j are arbitrary integers, and Δ is the resolution, i.e., the ~~minimal~~ unit of discretization. In drawing lines between grid points, this needs to be taken into account.

(better use "δ" rather than "Δ")

→ DOES NOT EXPLAIN HOW OBJECT 4 IN 395 CORY IS CONCAVE!

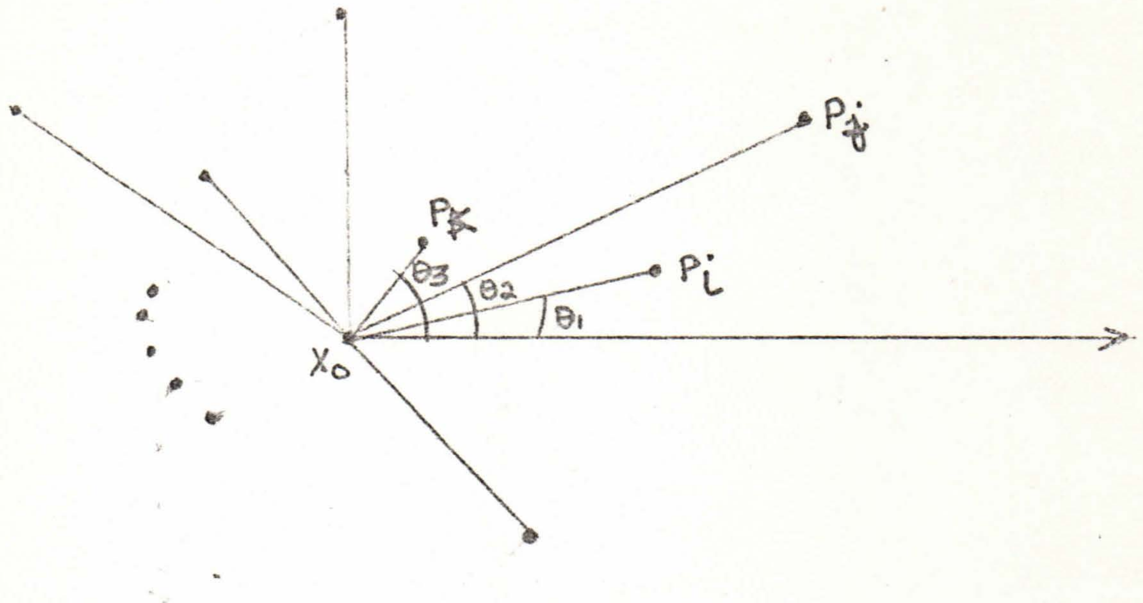


Fig. 2 - Ordering points by increasing angle

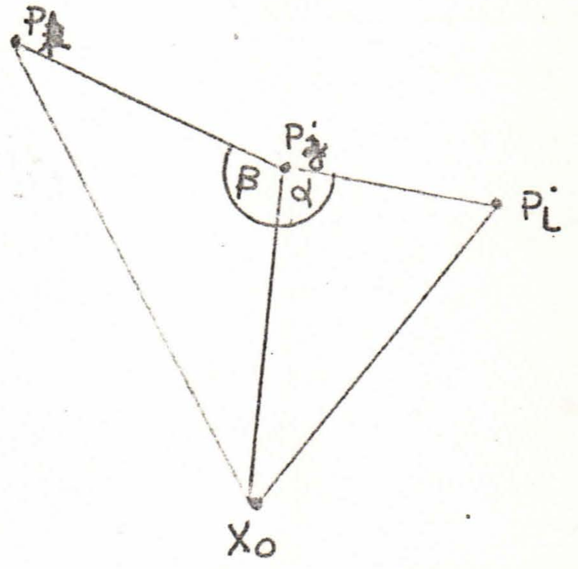


Fig. 3 - Testing a point for inclusion on the hull

In order to construct line segments whose points lie on lattice points of the grid, I used the following algorithm (my own):

<u>Algorithm for generating the points on the line segment Pi,Pj in discretized space</u>

```
1. Consider Pi = (Xi,Yi), Pj = (Xj,Yj)
2. If Xi<Xj set dx =Δ, else if Xi>Xj set dx =-Δ, else set dx = 0.
3. If Yi<Yj set dy =Δ, else if Yi>Yj set dy =-Δ, else set dy = 0.
4. Set Pk = Pi
5. Pk is the next point on the line segment.
6. If Pk = Pj, halt, all points (less Pj) have been generated
7. If Xk ≠ Xj, set Xk=Xk+dx, else set Xk = 0.
8. If Yk ≠ Yj, set Yk=Yk+dy, else set Yk = 0.
9. Go to step 5.
```

This algorithm may break the line segment into two line segments, (connected), unless all line segment points lie on lattice points. (see fig. 4)

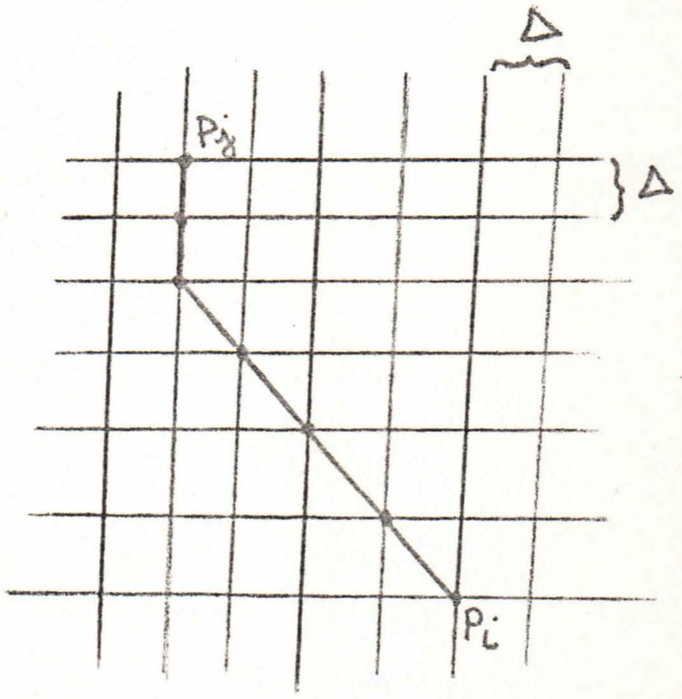


Fig. 4 - Line Segment Generation in Discrete Space

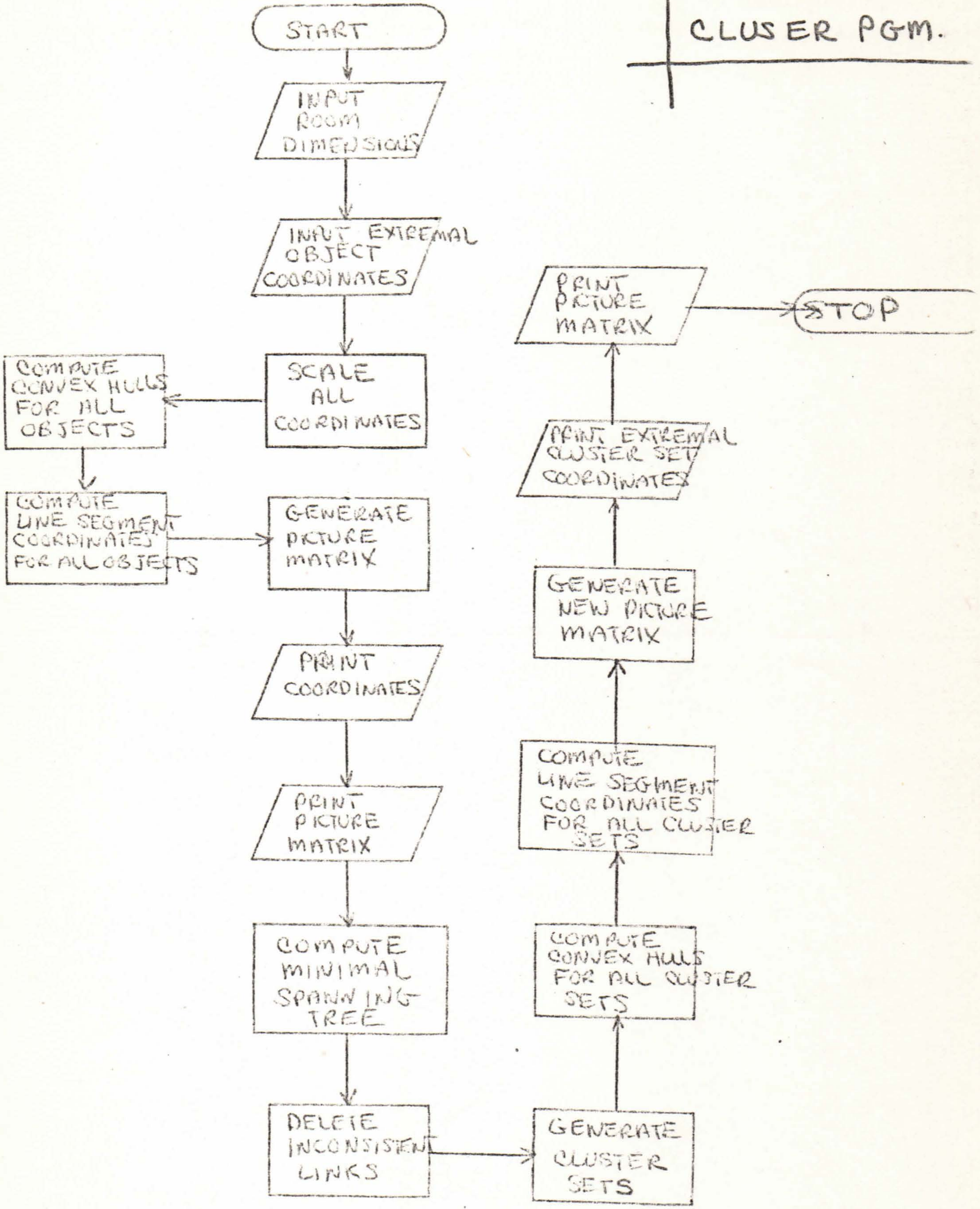
IV. Program Structure

The program as it exists now is designed to generate pseudo-objects in rooms up to 30' x 30' (this can be easily adjusted). Objects in the rooms may be of any shape and assume any orientation, as long as their extremal coordinates lie on lattice points defined by the resolution ~~minimal discretization~~ increment. (The value used for all test runs was $\Delta = .25'$). The hierarchy of program subroutines is rather straightforward and is outlined in the flowchart of Fig. 5.

While the program was written and debugged in APL, the full range of APL special operators and functions was purposely avoided ~~not made use of~~, in an effort to constrain algorithm implementation to a more Fortran~~-~~like style. I chose to write in APL rather than Fortran because of its interactive nature~~,~~ and interactive debugging facilities. I wanted to be able to alter parameters at various steps in the program~~,~~ while the program was running~~,~~ in order to enhance ~~the~~ program performance. In addition, the system is a fairly complex one, and I felt that I probably could not have gotten it debugged ~~completely working~~ in time, had I used Fortran from scratch.

I feel that the documentation which this paper and the program listings will provide, will enable a fairly simple conversion to Fortran, and subsequent incorporation into routines for Jason.

FLOW OF CONTROL FOR CLUSER PGM.



START
INPUT ROOM DIMENSIONS
INPUT EXTREMAL OBJECT COORDINATES
SCALE ALL COORDINATES
COMPUTE CONVEX HULLS FOR ALL OBJECTS
COMPUTE LINE SEGMENT COORDINATES FOR ALL OBJECTS
GENERATE PICTURE MATRIX
PRINT COORDINATES
PRINT PICTURE MATRIX
COMPUTE MINIMAL SPANNING TREE
DELETE INCONSISTENT LINKS
GENERATE CLUSTER SETS
COMPUTE CONVEX HULLS FOR ALL CLUSTER SETS
COMPUTE LINE SEGMENT COORDINATES FOR ALL CLUSTER SETS
GENERATE NEW PICTURE MATRIX
PRINT EXTREMAL CLUSTER SET COORDINATES
PRINT PICTURE MATRIX
STOP

Fig. 5.

V. Conclusion

The test runs made with the program on the three Cory Hall rooms (see Appendix II) were illustrative in pointing up the successes and faults of the approach used. Room 395 was a particularly good result, and the combination of the convex hull and the line drawing algorithm worked to advantage. However, in Room 293, pseudo-object #1 covers far more area than the objects which it encompasses. More investigation needs to be made of methods for object enclosure, once cluster sets have been determined. (I plan to add a feature to the program which will calculate free space in the room before and after clustering.)

The use of the minimal spanning tree concept was quite successful, but two significant difficulties arose. In the test run for Room 237 Cory, the fixed podium (object #35) and a desk (object #33) are not clustered together, despite the fact that they are physically quite close. This is due to the fact that distance is measured between midpoints, rather than extremal points. Performance would no doubt be improved by considering all extremal points of all objects in ✓ the generation of the spanning tree, but in addition to greatly increasing the amount of computation necessary, cluster set conflicts might very well arise, and would have to be dealt with.

The second difficulty with the method is in my opinion more significant, and has to do with the threshold chosen for inconsistent edge determination. The method as implemented is a one-shot clustering method, and performs no secondary clustering. (An example of sequential secondary clustering is given in Appendix II, using the Varied Object Room) This would not be very difficult to implement given the existing program, and is probably a good starting point for future work.

REFERENCES

See my bibliog for proper syntax of references.

REFERENCES

1. Sobek, R.P. and Sinclair, P.L., "The Preliminary Jason Reference Manual: Or What Makes Jason Run?," Department of Electrical Engineering and Computer Science, University of California, Berkeley, unpublished paper (June, 1974)

2. Cooper, G., Hankins, C. et al., "Jason the Berkeley Robot: Modeling and Problem Solving," Term project for EECS 290G, (Spring 1973).

3. Deo, N., Graph Theory With Applications to Engineering and Computer Science, (Prentice Hall Inc., Englewood Cliffs, New Jersey, 1974) Pp. 61-64

4. Warshall, S., "A Theorem on Boolean Matrices", JACM 9, (1962), Pp. 11-12 .

5. Zahn, C.T., "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters, IEEE Transactions on Computers, Vol. 1, (January 1971), Pp. 68-86.

6. Slagle, J.R., Chang, C.-L., and Lee, R.C.T., "Experiments With Some Cluster Analysis Algorithms", Pattern Recognition, Vol. 6, (June 1974), Pp. 181-187.

7. Slagle, J.R., Chang, C.-L., and Heller, S.R., "A Clustering and Data-Reorganizing Algorithm", IEEE Transactions on Systems, Man, and Cybernetics (January 1975), Pp. 125-128 .

8. Graham, R.L., "An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set", Information Processing Letters, Vol. 1, January 1972, Pp. 132-133.

9. Jarvis, R.A., "On the Identification of the Convex Hull of a Finite Set of Points in the Plane", Information Processing Letters, Vol. 2, February 1973, Pp. 18-21.

10. Potts, R.B. and Oliver, R.M., Flows in Transportation Networks, Academic Press, New York, New York, 1972, pp. 49-67.

APPENDICES I - IV

ROOM 237 CORY HALL

Vertices of room : (0,23) (19.5,23) (0,0) (19.5,0)

Door 237 : (13.5,0) (17,0)

<u>OBJECTS IN THE ROOM</u> :

1. Desk (.25,20.5)(2.5,20.5)(2.5,22.5)(.25,22.5)
2. Desk (.25,18)(2.5,18)(2.5,20)(.25,20)
3. Desk (.25,15.75)(2.5,15.75)(2.5,27.75)(.25,17.75)
4. Desk (.25,13.5)(2.5,13.5)(2.5,15.5)(.25,15.5)
5. Desk (3,20.5)(5.25,20.5)(5.25,22.5)(3,22.5)
6. Desk (3,18)(5.25,18)(5.25,20)(3,20)
7. Desk (3,15.75)(5.25,15.75)(5.25,17.75)(3,17.75)
8. Desk (3,13.5)(5.25,13.5)(5.25,15.5)(3,15.5)
9. Desk (5.75,20.5)(8,20.5)(8,22.5)(5.75,22.5)
10. Desk (5.75,20.25)(8,20.25)(8,18.25)(5.75,18.25)
11. Desk (5.75,16)(8,16)(8,18)(5.75,18)
12. Desk (5.75,13.75)(8,13.75)(8,15.75)(5.75,15.75)
13. Desk (8.5,22.5)(10.75,22.5)(10.75,20.5)(8.5,20.5)
14. Desk (8.5,20.25)(10.75,20.25)(10.75,18.25)(8.5,18.25)
15. Desk (8.5,18)(10.75,18)(10.75,16)(8.5,16)
16. Desk (8.5,15.75)(10.75,15.75)(10.75,13.75)(8.5,13.75)
17. Desk (.25,9)(2.5,9)(2.5,7)(.25,7)
18. Desk (.25,6.75)(2.5,6.75)(2.5,4.75)(.25,4.75)
19. Desk (.25,4.5)(2.5,4.5)(2.5,2.5)(.25,2.5)
20. Desk (.25,2.25)(2.5,2.25)(2.5,.25)(.25,.25)
21. Desk (3,9)(5.25,9)(5.25,7)(3,7)
22. Desk (3,6.75)(5.25,6.75)(5.25,4.75)(3,4.75)
23. Desk (3,4.5)(5.25,4.5)(5.25,2.5)(3,2.5)

24. Desk (3,2.25)(5.25,2.25)(5.25,.25)(3,.25)

25. Desk (5.75,9)(8,9)(8,7)(5.75,7)

26. Desk (5.75,6.75)(8,6.75)(8,4.75)(5.75,4.75)

27. Desk (5.75,4.5)(8,4.5)(8,2.5)(5.75,2.5)

28. Desk (5.75,2.25)(8,2.25)(8,.25)(5.75,.25)

29. Desk (8.5,9.75)(10.75,9.75)(10.75,7.75)(8.5,7.75)

30. Desk (8.5,7.25)(10.75,7.25)(10.75,5.25)(8.5,5.25)

31. Desk (8.5,5)(10.75,5)(10.75,3)(8.5,3)

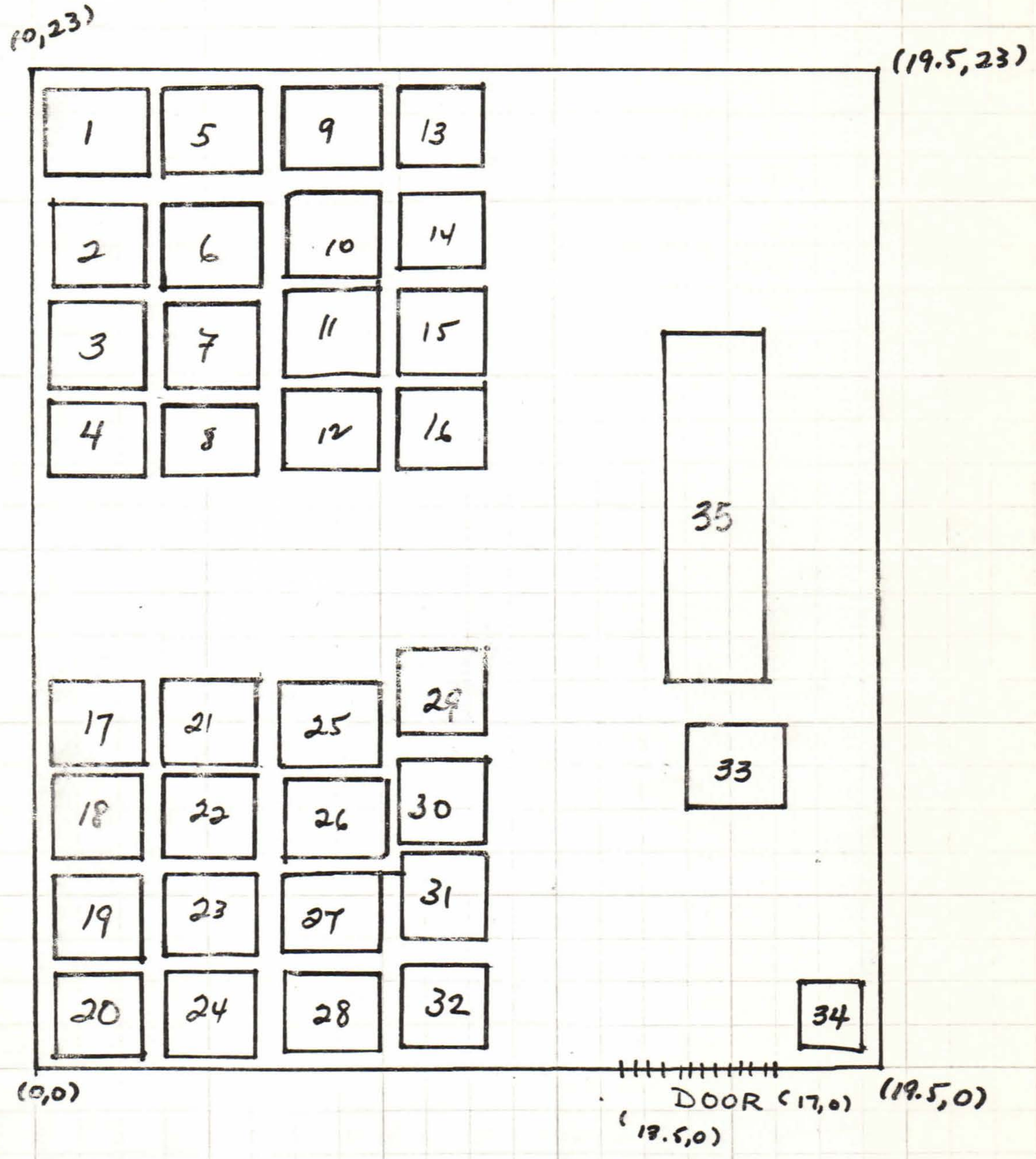
32. Desk (8.5,2.5)(10.75,2.5)(10.75,.5)(8.5,.5)

33. Desk (15,8)(17.25,8)(17.25,6)(15,6)

34. ~~Garbage~~ Wastepaper can (17.5,2)(19,2)(19,.5)(17.5,.5)

35. Fixed podium (14.5,9)(17.25,9)(17.25,17)(14.5,17)

Room 237 Cory



1 FOOT = 1 SQUARE

ROOM 293 CORY HALL

Vertices of room : (0,0)(20.5,0)(20.5,23)(0,23)

Door 293 : (2.25,23)(5.75,23)

OBJECTS IN THE ROOM :

1. desk (6.25,1)(8.5,1)(8.5,3)(6.25,3)
2. desk (9.25,20.5)(11.5,20.5)(11.5,22.5)(9.25,22.5)
3. desk (9.25,6)(11.5,6)(11.5,8)(9.25,8)
4. desk (9.5,3.5)(11.75,3.5)(11.75,5.5)(9.5,5.5)
5. desk (9.25,1)(11.5,1)(11.5,3)(9.25,3)
6. desk (12,20.5)(14.25,20.5)(14.25,22.5)(12,22.5)
7. desk (12,18.25)(14.25,18.25)(14.25,20.25)(12,20.25)
8. desk (12,16)(14.25,16)(14.25,18)(12,18)
9. desk (12,13.75)(14.25,13.75)(14.25,15.75)(12,15.75)
10. desk (12,11)(14.25,11)(14.25,13)(12,13)
11. desk (12,8.75)(14.25,8.75)(14.25,10.75)(12.10.75)
12. desk (12,6.5)(14.25,6.5)(14.25,8.5)(12,8.5)
13. desk (12,4)(14.25,4)(14.25,6)(12,6)
14. desk (12,1)(14.25,1)(14.25,3)(12,3)
15. desk (14.75,20.75)(17,20.75)(17,22.75)(14.75,22.75)
16. desk (14.75,18.5)(17,18.5)(17,20.5)(14.75,20.5)
17. desk (14.75,16.25)(17,16.25)(17,18.25)(14.75,18.25)
18. desk (14.75,14)(17,14)(17,16)(14.75,16)
19. desk (14.75,11.75)(17,11.75)(17,13.75)(14.75,13.75)
20. desk (14.75,9.25)(17,9.25)(17,11.25)(14.75,11.25)
21. desk (14.75,7)(17,7)(17,9)(14.75,9)
22. desk (14.75,6.75)(17,6.75)(17,4.75)(14.75,4.75)
23. desk (14.75,4.5)(17,4.5)(17,2.5)(14.75,2.5)
24. desk (14.75,2.25)(17,2.25)(17,.25)(14.75,.25)
25. desk (17.75,20.75)(20,20.75)(20,22.75)(17.75,22.75)

26. desk (17.75,18.5)(20,18.5)(20,20.5)(17.75,20.5)

27. desk (17.75,16.25)(20,16.25)(20,18.25)(17.75,18.25)

28. desk (17.75,14)(20,14)(20,16)(17.75,16)

29. desk (17.75,11.75)(20,11.75)(20,13.75)(17.75,13.75)

30. desk (17.75,9.5)(20,9.5)(20,11.5)(17.75,11.5)

31. desk (17.75,7.25)(20,7.25)(20,9.25)(17.75,9.25)

32. desk (17.75,5)(20,5)(20,7)(17.75,7)

33. desk (17.75,2.75)(20,2.75)(20,4.75)(17.75,4.75)

34. desk (17.75,.5)(20,.5)(20,2.5)(17.75,2.5)

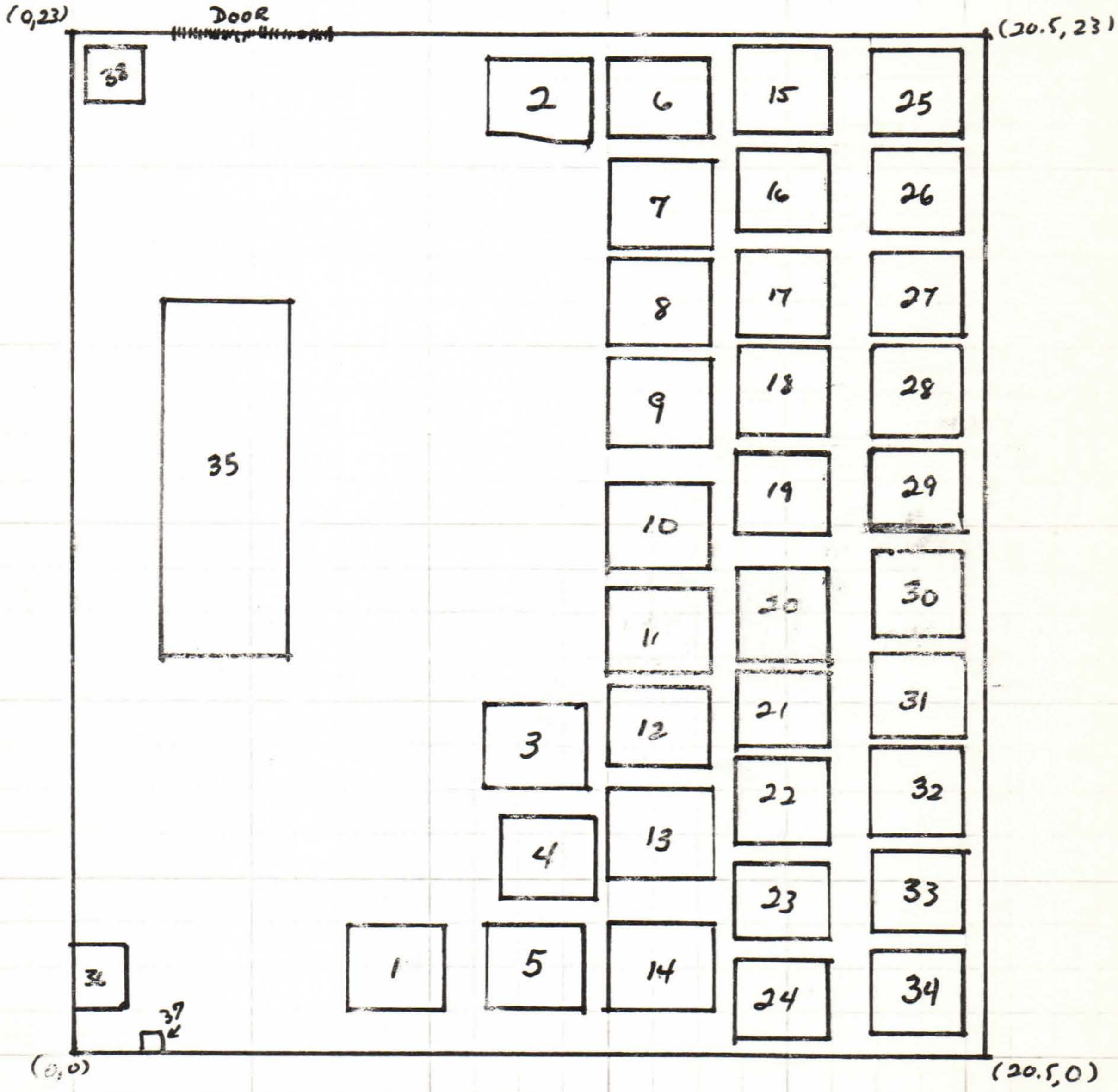
35. fixed podium (2,9)(5,9)(5,17)(2,17)

36. podium (0,1)(1.25,1)(1.25,2.5)(0,2.5)

37. pipe (1.5,0)(2,0)(2,.5)(1.5,.5)

38. waste paper ~~garbage~~ can (.25,21.5)(1.5,21.5)(1.5,22.75)(.25,22.75)

Room 293 Cory



1 FOOT = 1 SQUARE

ROOM 395 CORY HALL

Vertices of the room : (0,23)(0,0)(16.5,0)(16.5,23)

Door 395A : (3.5,23)(7,23)

Door 395B : (0,18.3)(0,21)

OBJECTS IN THE ROOM :

1. table (10,19)(13,19)(10,13)(13,13)
2. table (10,13)(13,13)(10,7)(13,7)
3. garbage can (8,21)(8,22.5)(9.5,21)(9.5,22.5)
4. chair (.5,21)(.5,22.5)(2,21)(2,22.5)
5. desk (.5,17)(.5,15.25)(2.5,17)(2.5,15.25)
6. desk (.5,15)(.5,13.25)(2.5,15)(2.5,13.25)
7. desk (.5,12.75)(.5,11)(2.5,12.75)(2.5,11)
8. desk (.5,10.75)(.5,9)(2.5,10.75)(2.5,9)
9. desk (.5,8.75)(.5,7)(2.5,8.75)(2.5,7)
10. desk (.5,6.75)(.5,5)(2.5,6.75)(2.5,5)
11. desk (.5,4.75)(.5,3)(2.5,4.75)(2.5,3)
12. desk (.5,2.75)(.5,1)(2.5,2.75)(2.5,1)
13. desk (3,17)(5,17)(5,15.25)(3,15.25)
14. desk (3,15)(5,15)(5,13.25)(3,13.25)
15. desk (3,13)(5,13)(5,11.25)(3,11.25)
16. desk (3,9.25)(5,9.25)(5,7.5)(3,7.5)
17. chair (3,11)(4.5,11)(4.5,9.5)(3,9.5)
18. chair (3,7.25)(4.5,7.25)(4.5,5.75)(3,5.75)
19. chair (3,5.5)(4.5,5.5)(4.5,4)(3,4)
20. chair (3,3.75)(4.5,3.75)(4.5,2.25)(3,2.25)
21. chair (3,2)`4.5,2)(4.5,.5)(2,.5)
22. chair (5,6.5)(6.5,6.5)(6.5,5)(5,5)

23. chair (5,4.5)(6.5,4.5)(6.5,3)(5,3)

24. chair (5,2.5)(6.5,2.5)(6.5,1)(5,1)

25. chair (7,5.5)(8.5,5.5)(8.5,4)(7,4)

26. chair (7,3.75)(8.5,3.75)(8.5,2.25)(7,2.25)

27. chair (7,2)(8.5,2)(8.5,.5)(7,.5)

28. chair (9,2)(10.5,2)(10.5,.5)(9,.5)

29. chair (11,2)(12.5,2)(12.5,.5)(11,.5)

30. chair (9.5,18)(10,18)(10,16.5)(9.5,16.5)

31. chair (9.5,16)(10,16)(10,14.5)(9.5,14.5)

32. chair (9.5,13.5)(10,13.5)(10,12)(9.5,12)

33. chair (9.5,11.5)(10,11.5)(10,10)(9.5,10)

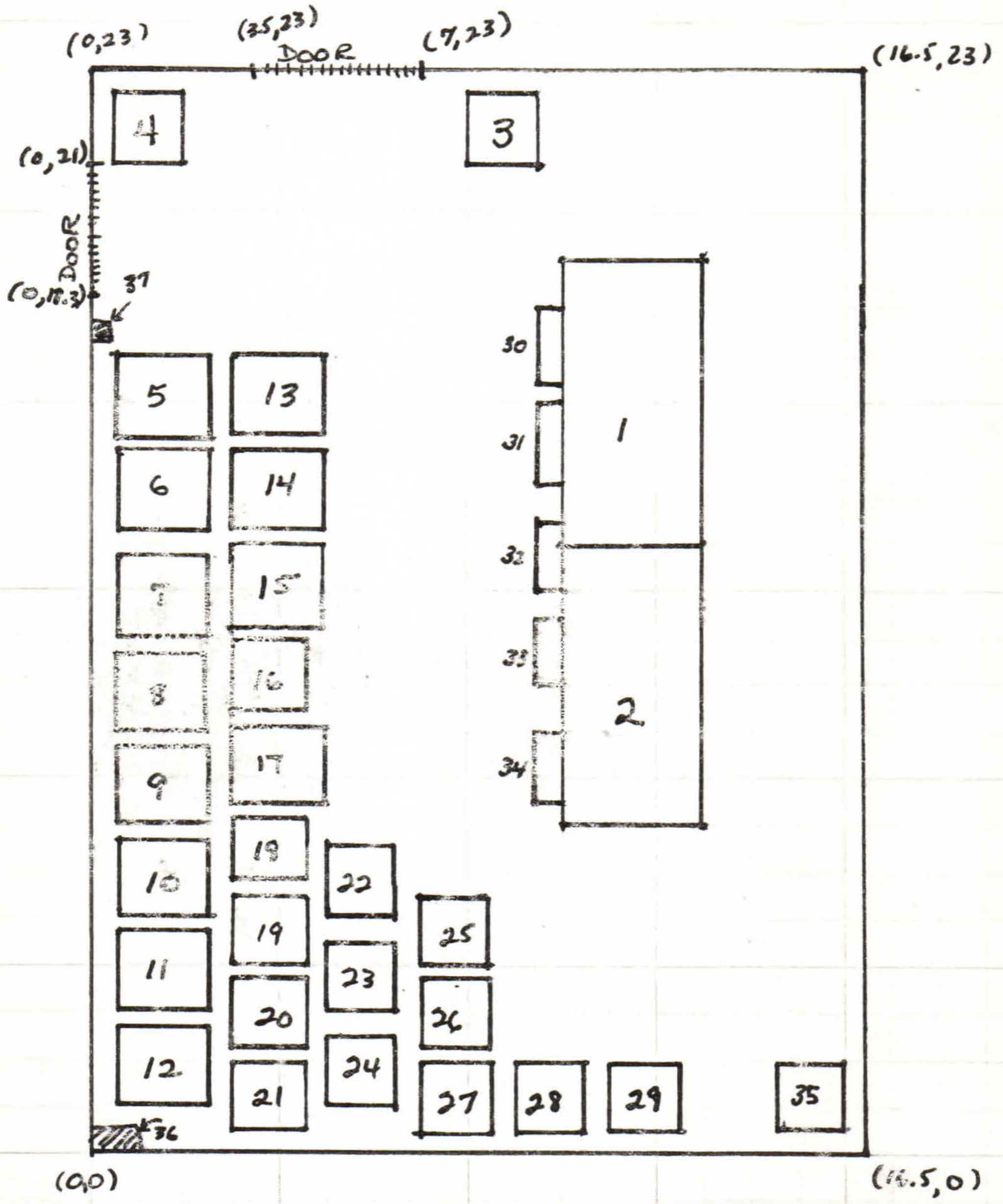
34. chair (9.5,9)(10,9)(10,7.5)(9.5,7.5)

35. waste paper ~~garbage~~ can (14.5,.5)(16,.5)(16,2)(14.5,2)

36. pipe (0,0)(1,0)(1,.5)(0,.5)

37. pipe (0,17.25)(.5,17.25)(.5,17.75)(0,17.75)

Room 395 Cory



1 FOOT = 1 SQUARE

237
CORY

```
CLUSTER

TYPE IN NAME OF ROOM
237 CORY

INPUT THE ROOM COORDINATES
⎕:
      0 23 19.5 23 19.5 0 0 0

INPUT SCALE INCREMENT FACTOR
⎕:
      .25

INPUT OBJECT COORDINATES, ONE SET PER LINE.  HIT 0 AFTER LAST SET.
⎕:
      .25 20.5 2.5 20.5 2.5 22.5 .25 22.5
⎕:
      .25 18 2.5 18 2.5 20 .25 20
⎕:
      .25 15.75 2.5 15.75 2.5 17.75 .25 17.75
⎕:
      .25 13.5 2.5 13.5 2.5 15.5 .25 15.5
⎕:
      3 20.5 5.25 20.5 5.25 22.5 3 22.5
⎕:
      3 18 5.25 18 5.25 20 3 20
⎕:
      3 15.75 5.25 15.75 5.25 17.75 3 17.75
⎕:
      3 13.5 5.25 13.5 5.25 15.5 3 15.5
⎕:
      5.75 20.5 8 20.5 8 22.5 5.75 22.5
⎕:
      5.75 20.25 8 20.25 8 18.25 5.75 18.25
⎕:
      5.75 16 8 16 8 18 5.75 18
⎕:
      5.75 13.75 8 13.75 8 15.75 5.75 15.75
⎕:
      8.5 22.5 10.75 22.5 10.75 20.5 8.5 20.5
⎕:
      8.5 20.25 10.75 20.25 10.75 18.25 8.5 18.25
⎕:
      8.5 18 10.75 18 10.75 16 8.5 16
⎕:
      8.5 15.75 10.75 15.75 10.75 13.75 8.5 13.75
⎕:
      .25 9 2.5 9 2.5 7 .25 7
⎕:
      .25 6.75 2.5 6.75 2.5 4.75 .25 4.75
⎕:
      .25 4.5 2.5 4.5 2.5 2.5 .25 2.5
⎕:
      .25 2.25 2.5 2.25 2.5 .25 .25 .25
⎕:
      3 9 5.25 9 5.25 7 3 7
⎕:
      3 6.75 5.25 6.75 5.25 4.75 3 4.75
⎕:
      3 4.5 5.25 4.5 5.25 2.5 3 2.5
⎕:
      3 2.25 5.25 2.25 5.25 .25 3 .25
⎕:
      5.75 9 8 9 8 7 5.75 7

⎕:
      5.75 6.75 8 6.75 8 4.75 5.75 4.75
⎕:
      5.75 4.5 8 4.5 8 2.5 5.75 2.5
⎕:
      5.75 2.25 8 2.25 8 .25 5.75 .25
⎕:
      8.5 9.75 10.75 9.75 10.75 7.75 8.5 7.75
⎕:
      8.5 7.25 10.75 7.25 10.75 5.25 8.5 5.25
⎕:
      8.5 5 10.75 5 10.75 3 8.5 3
⎕:
      8.5 2.5 10.75 2.5 10.75 .5 8.5 .5
⎕:
      15 8 17.25 8 17.25 6 15 6
⎕:
      17.5 2 19 2 19 .5 17.5 .5
⎕:
      14.5 9 17.25 9 17.25 17 14.5 17
⎕:
      0
```

237
CORY

EXTREMAL OBJECT COORDINATES

POINT	X	Y
1	0.25	20.5
2	2.5	20.5
3	2.5	22.5
4	0.25	22.5
5	0.25	18
6	2.5	18
7	2.5	20
8	0.25	20
9	0.25	15.75
10	2.5	15.75
11	2.5	17.75
12	0.25	17.75
13	0.25	13.5
14	2.5	13.5
15	2.5	15.5
16	0.25	15.5
17	3	20.5
18	5.25	20.5
19	5.25	22.5
20	3	22.5
21	3	18
22	5.25	18
23	5.25	20
24	3	20
25	3	15.75
26	5.25	15.75
27	5.25	17.75
28	3	17.75
29	3	13.5
30	5.25	13.5
31	5.25	15.5
32	3	15.5
33	5.75	20.5
34	8	20.5
35	8	22.5
36	5.75	22.5
37	5.75	20.25
38	8	20.25
39	8	18.25
40	5.75	18.25
41	5.75	16
42	8	16
43	8	18
44	5.75	18
45	5.75	13.75
46	8	13.75
47	8	15.75
48	5.75	15.75
49	8.5	22.5
50	10.75	22.5
51	10.75	20.5
52	8.5	20.5
53	8.5	20.25
54	10.75	20.25
55	10.75	18.25
56	8.5	18.25
57	8.5	18

237
CORY

```
 58          10.75      16
 59          10.75      16
 60           8.5       16
 61           8.5       15.75
 62          10.75      15.75
 63          10.75      13.75
 64           8.5       13.75
 65           0.25       9
 66           2.5        9
 67           2.5        7
 68           0.25       7
 69           0.25       6.75
 70           2.5        6.75
 71           2.5        4.75
 72           0.25       4.75
 73           0.25       4.5
 74           2.5        4.5
 75           2.5        2.5
 76           0.25       2.5
 77           0.25       2.25
 78           2.5        2.25
 79           2.5        0.25
 80           0.25       0.25
 81           3          9
 82           5.25       9
 83           5.25       7
 84           3          7
 85           3          6.75
 86           5.25       6.75
 87           5.25       4.75
 88           3          4.75
 89           3          4.5
 90           5.25       4.5
 91           5.25       2.5
 92           3          2.5
 93           3          2.25
 94           5.25       2.25
 95           5.25       0.25
 96           3          0.25
 97           5.75       9
 98           8          9
 99           8          7
100           5.75       7
101           5.75       6.75
102           8          6.75
103           8          4.75
104           5.75       4.75
105           5.75       4.5
106           8          4.5
107           8          2.5
108           5.75       2.5
109           5.75       2.25
110           8          2.25
111           8          0.25
112           5.75       0.25
113           8.5        9.75
114          10.75       9.75
115          10.75       7.75
116           8.5        7.75
117           8.5        7.25
118          10.75       7.25
119          10.75       5.25
120           8.5        5.25
121           8.5        5
122          10.75       5
123          10.75       3
```

237
CORY

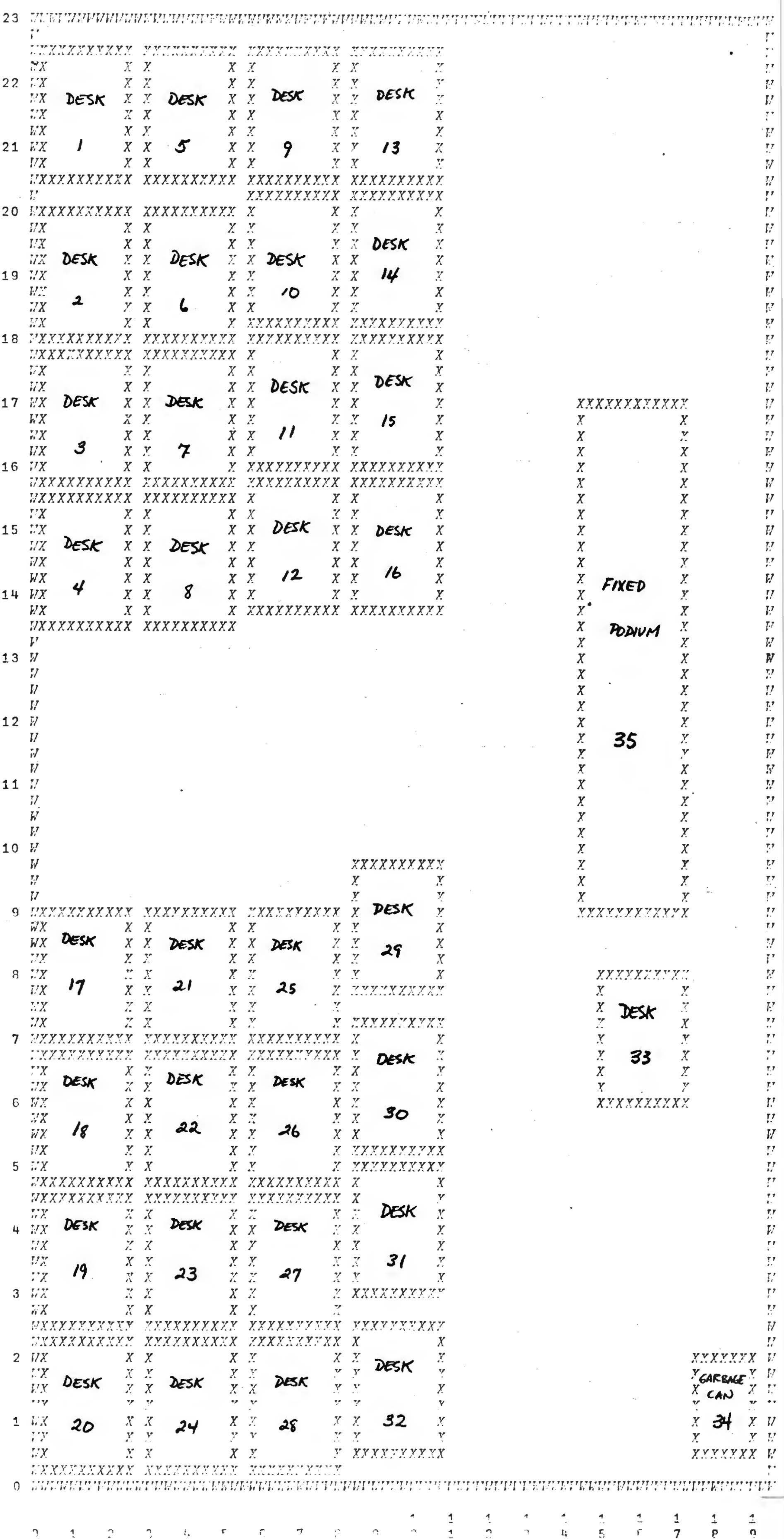
124	8.5	3
125	8.5	2.5
126	10.75	2.5
127	10.75	0.5
128	8.5	0.5
129	15	8
130	17.25	8
131	17.25	6
132	15	6
133	17.5	2
134	19	2
135	19	0.5
136	17.5	0.5
137	14.5	9
138	17.25	9
139	17.25	17
140	14.5	17

OBJ.	P1	PN
1	1	4
2	5	8
3	9	12
4	13	16
5	17	20
6	21	24
7	25	28
8	29	32
9	33	36
10	37	40
11	41	44
12	45	48
13	49	52
14	53	56
15	57	60
16	61	64
17	65	68
18	69	72
19	73	76
20	77	80
21	81	84
22	85	88
23	89	92
24	93	96
25	97	100
26	101	104
27	105	108
28	109	112
29	113	116
30	117	120
31	121	124
32	125	128
33	129	132
34	133	136
35	137	140

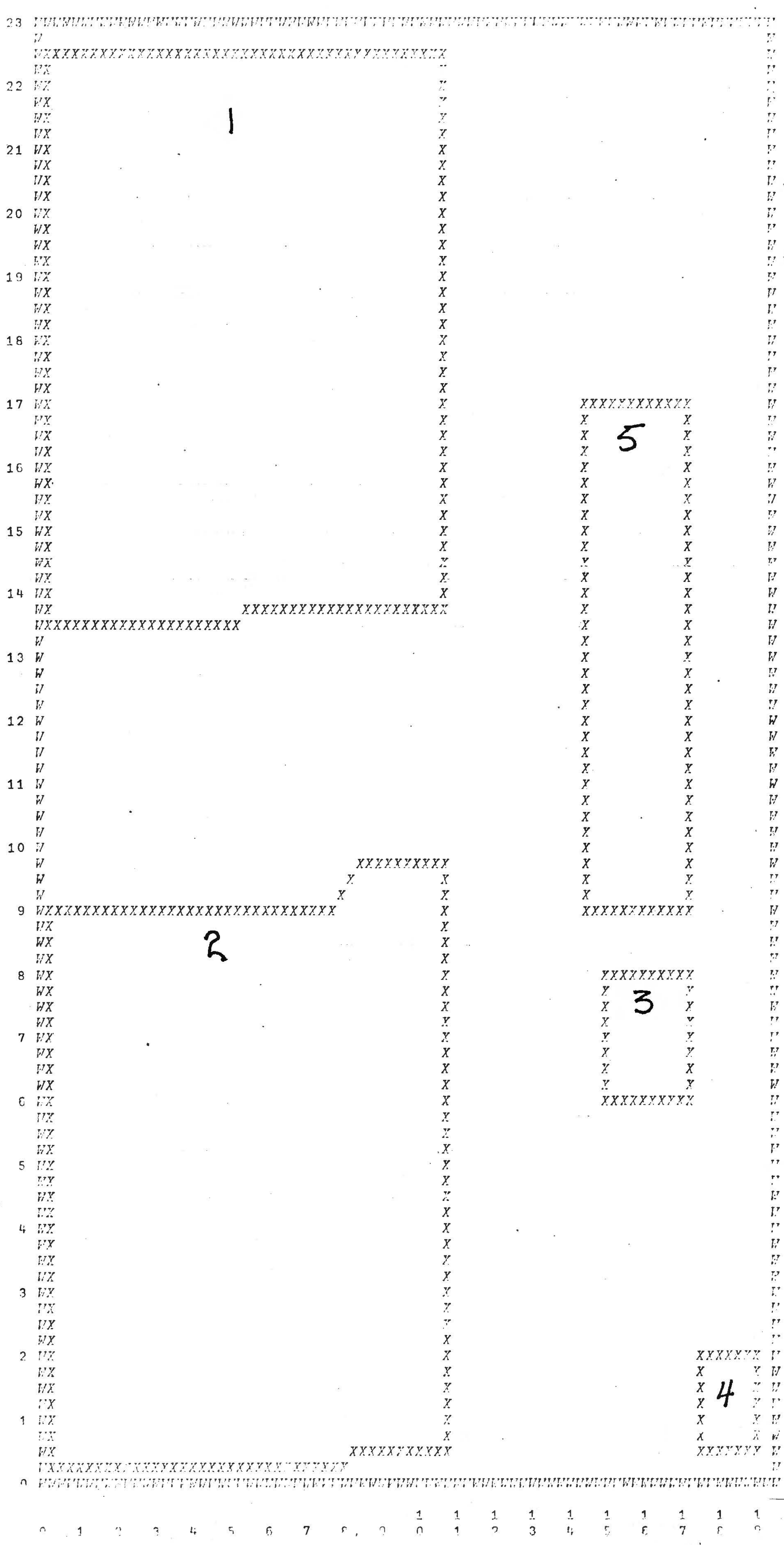
MOVE CARRIAGE TO TOP OF NEW PAGE. HIT CARRIAGE RETURN

```
WEIGHT OF MIN. SPANNING TREE = 96.14
AVG. LINK LENGTH             = 2.828
S.D. OF LINKS                =1.232

LINK NO.      LINK(VI→VJ)          LINK WT.       WT./AVG.   DEV.
--------      -----------          --------       --------   ----

   1          5          6           2.5          0.8842     ¯0.3276
   2          6          7           2.25         0.7957     ¯0.5776
   3          7          8           2.25         0.7957     ¯0.5776
   4          5          1           2.75         0.9726     ¯0.07756
   5          1          2           2.5          0.8842     ¯0.3276
   6          2          3           2.25         0.7957     ¯0.5776
   7          3          4           2.25         0.7957     ¯0.5776
   8          5          9           2.75         0.9726     ¯0.07756
   9          9         10           2.25         0.7957     ¯0.5776
  10         10         11           2.25         0.7957     ¯0.5776
  11         11         12           2.25         0.7957     ¯0.5776
  12          9         13           2.75         0.9726     ¯0.07756
  13         13         14           2.25         0.7957     ¯0.5776
  14         14         15           2.25         0.7957     ¯0.5776
  15         15         16           2.25         0.7957     ¯0.5776
  16         16         29           6            2.122       3.172 x
  17         29         30           2.5          0.8842     ¯0.3276
  18         30         31           2.25         0.7957     ¯0.5776
  19         31         32           2.5          0.8842     ¯0.3276
  20         32         28           2.761        0.9766     ¯0.06622
  21         28         27           2.25         0.7957     ¯0.5776
  22         27         26           2.25         0.7957     ¯0.5776
  23         26         25           2.25         0.7957     ¯0.5776
  24         28         24           2.75         0.9726     ¯0.07756
  25         24         23           2.25         0.7957     ¯0.5776
  26         23         22           2.25         0.7957     ¯0.5776
  27         22         21           2.25         0.7957     ¯0.5776
  28         24         20           2.75         0.9726     ¯0.07756
  29         20         19           2.25         0.7957     ¯0.5776
  30         19         18           2.25         0.7957     ¯0.5776
  31         18         17           2.25         0.7957     ¯0.5776
  32         16         35           6.49         2.295       3.663 x
  33         35         33           6.005        2.124       3.178 x
  34         33         34           6.13         2.168       3.303 x

INPUT EDGE INCONSISTENCY FACTOR (NO. STD. DEVS.)
⎕:
      1


  INCONSISTENT LINKS:  16  32  33  34

CLUSTER NO. 1:  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16

CLUSTER NO. 2:  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32

   STER NO. 3:  33

CLUSTER NO. 4:  34

CLUSTER NO. 5:  35
```

237 CORY - UNCLUTTERED



237 CORY CLUSTERED



237 CORY

EXTREMAL PSEUDO-OBJECT COORDINATES

POINT	X	Y
1	10.75	18
2	10.75	18.25
3	10.75	20.25
4	10.75	20.5
5	10.75	22.5
6	8.5	22.5
7	8	22.5
8	5.75	22.5
9	5.25	22.5
10	3	22.5
11	2.5	22.5
12	0.25	22.5
13	0.25	20.5
14	0.25	20
15	0.25	18
16	0.25	17.75
17	0.25	15.75
18	0.25	15.5
19	0.25	13.5
20	2.5	13.5
21	3	13.5
22	5.25	13.5
23	10.75	13.75
24	10.75	15.75
25	10.75	16
26	10.75	5
27	10.75	5.25
28	10.75	7.25
29	10.75	7.75
30	10.75	9.75
31	8.5	9.75
32	0.25	9
33	0.25	7
34	0.25	6.75
35	0.25	4.75
36	0.25	4.5
37	0.25	2.5
38	0.25	2.25
39	0.25	0.25
40	2.5	0.25
41	3	0.25
42	5.25	0.25
43	5.75	0.25
44	8	0.25
45	10.75	0.5
46	10.75	2.5
47	10.75	3
48	17.25	8
49	15	8
50	15	6
51	17.25	6
52	19	2
53	17.5	2
54	17.5	0.5
55	19	0.5
56	17.25	17
57	14.5	17
58	14.5	9
59	17.25	9

OBJ.	P1	PN
1	1	25
2	26	47
3	48	51
4	52	55
5	56	59

(38)

293
CORY

(38)

```
        CLUSTER
TYPE IN NAME OF ROOM
293 CORY

INPUT THE ROOM COORDINATES
⎕:
        0 0 20.5 0 20.5 23 0 23

INPUT SCALE INCREMENT FACTOR
⎕:
        .25

INPUT OBJECT COORDINATES, ONE SET PER LINE.  HIT 0 AFTER LAST SET.
⎕:
        6.25 1 8.5 1 8.5 3 6.25 3
⎕:
        9.25 20.5 11.5 20.5 11.5 22.5 9.25 22.5
⎕:
        9.25 6 11.5 6 11.5 8 9.25 8
⎕:
        9.5 3.5 11.75 3.5 11.75 5.5 9.5 5.5
⎕:
        9.25 1 11.5 1 11.5 3 9.25 3
⎕:
        12 20.5 14.25 20.5 14.25 22.5 12 22.5
⎕:
        12 18.25 14.25 18.25 14.25 20.25 12 20.25
⎕:
        12 16 14.25 16 14.25 18 12 18
⎕:
        12 13.75 14.25 13.75 14.25 15.75 12 15.75
⎕:
        12 11 14.25 11 14.25 13 12 13
⎕:
        12 8.75 14.25 8.75 14.25 10.75 12 10.75
⎕:
        12 6.5 14.25 6.5 14.25 8.5 12 8.5
⎕:
        12 4 14.25 4 14.25 6 12 6
⎕:
        12 1 14.25 1 14.25 3 12 3
⎕:
        14.75 20.75 17 20.75 17 22.75 14.75 22.75
⎕:
        14.75 18.5 17 18.5 17 20.5 14.75 20.5
⎕:
        14.75 16.25 17 16.25 17 18.25 14.75 18.25
⎕:
        14.75 14 17 14 17 16 14.75 16
⎕:
        14.75 11.75 17 11.75 17 13.75 14.75 13.75
⎕:
        14.75 9.25 17 9.25 17 11.25 14.75 11.25
⎕:
        14.75 7 17 7 17 9 14.75 9
⎕:
        14.75 6.75 17 6.75 17 4.75 14.75 4.75
⎕:
        14.75 4.5 17 4.5 17 2.5 14.75 2.5
⎕:
        14.75 2.25 17 2.25 17 .25 14.75 .25
⎕:
        17.75 20.75 20 20.75 20 22.75 17.75 22.75
⎕:

        17.75 18.5 20 18.5 20 20.5 17.75 20.5
⎕:
        17.75 16.25 20 16.25 20 18.25 17.75 18.25
⎕:
        17.75 14 20 14 20 16 17.75 16 .
⎕:
        17.75 11.75 20 11.75 20 13.75 17.75 13.75
⎕:
        17.75 9.5 20 9.5 20 11.5 17.75 11.5
⎕:
        17.75 7.25 20 7.25 20 9.25 17.75 9.25
⎕:
        17.75 5 20 5 20 7 17.75 7
⎕:
        17.75 2.75 20 2.75 20 4.75 17.75 4.75
⎕:
        17.75 .5 20 .5 20 2.5 17.75 2.5
⎕:
        2 9 5 9 5 17 2 17
⎕:
        0 1 1.25 1 1.25 2.5 0 2.5
⎕:
        1.5 0 2 0 2 .5 1.5 .5
⎕:
        .25 21.5 1.5 21.5 1.5 22.75 .25 22.75
⎕:
        0
```

EXTREMAL OBJECT COORDINATES

293
CORY

POINT	X	Y
1	6.25	1
2	8.5	1
3	8.5	3
4	6.25	3
5	9.25	20.5
6	11.5	20.5
7	11.5	22.5
8	9.25	22.5
9	9.25	6
10	11.5	6
11	11.5	8
12	9.25	8
13	9.5	3.5
14	11.75	3.5
15	11.75	5.5
16	9.5	5.5
17	9.25	1
18	11.5	1
19	11.5	3
20	9.25	3
21	12	20.5
22	14.25	20.5
23	14.25	22.5
24	12	22.5
25	12	18.25
26	14.25	18.25
27	14.25	20.25
28	12	20.25
29	12	16
30	14.25	16
31	14.25	18
32	12	18
33	12	13.75
34	14.25	13.75
35	14.25	15.75
36	12	15.75
37	12	11
38	14.25	11
39	14.25	13
40	12	13
41	12	8.75
42	14.25	8.75
43	14.25	10.75
44	12	10.75
45	12	6.5
46	14.25	6.5
47	14.25	8.5
48	12	8.5
49	12	4
50	14.25	4
51	14.25	6
52	12	6
53	12	1
54	14.25	1
55	14.25	3
56	12	3

293
CORY

57	14.75	20.75
58	17	20.75
59	17	22.75
60	14.75	22.75
61	14.75	18.5
62	17	18.5
63	17	20.5
64	14.75	20.5
65	14.75	16.25
66	17	16.25
67	17	18.25
68	14.75	18.25
69	14.75	14
70	17	14
71	17	16
72	14.75	16
73	14.75	11.75
74	17	11.75
75	17	13.75
76	14.75	13.75
77	14.75	9.25
78	17	9.25
79	17	11.25
80	14.75	11.25
81	14.75	7
82	17	7
83	17	9
84	14.75	9
85	14.75	6.75
86	17	6.75
87	17	4.75
88	14.75	4.75
89	14.75	4.5
90	17	4.5
91	17	2.5
92	14.75	2.5
93	14.75	2.25
94	17	2.25
95	17	0.25
96	14.75	0.25
97	17.75	20.75
98	20	20.75
99	20	22.75
100	17.75	22.75
101	17.75	18.5
102	20	18.5
103	20	20.5
104	17.75	20.5
105	17.75	16.25
106	20	16.25
107	20	18.25
108	17.75	18.25
109	17.75	14
110	20	14
111	20	16
112	17.75	16
113	17.75	11.75
114	20	11.75
115	20	13.75
116	17.75	13.75
117	17.75	9.5
118	20	9.5
119	20	11.5
120	17.75	11.5
121	17.75	7.25

293
CORY

123	20	9.25
124	17.75	9.25
125	17.75	5
126	20	5
127	20	7
128	17.75	7
129	17.75	2.75
130	20	2.75
131	20	4.75
132	17.75	4.75
133	17.75	0.5
134	20	0.5
135	20	2.5
136	17.75	2.5
137	2	9
138	5	9
139	5	17
140	2	17
141	0	1
142	1.25	1
143	1.25	2.5
144	0	2.5
145	1.5	0
146	2	0
147	2	0.5
148	1.5	0.5
149	0.25	21.5
150	1.5	21.5
151	1.5	22.75
152	0.25	22.75

OBJ.	*P1*	*PN*
1	1	4
2	5	8
3	9	12
4	13	16
5	17	20
6	21	24
7	25	28
8	29	32
9	33	36
10	37	40
11	41	44
12	45	48
13	49	52
14	53	56
15	57	60
16	61	64
17	65	68
18	69	72
19	73	76
20	77	80
21	81	84
22	85	88
23	89	92
24	93	96
25	97	100
26	101	104
27	105	108
28	109	112
29	113	116
30	117	120
31	121	124
32	125	128
33	129	132
34	133	136
35	137	140
36	141	144
37	145	148
38	149	152

```
WEIGHT OF MIN. SPANNING TREE  = 105.5
AVG. LINK LENGTH              = 2.851
S.D. OF LINKS                 =1.663

LINK NO.        LINK(VI→VJ)         LINK WT.      WT./AVG.   DEV.
--------        -----------         --------      --------   ----

    1               5          4      2.512         0.8814     ¯0.3381
    2               4          3      2.512         0.8814     ¯0.3381
    3               4         13      2.55          0.8944     ¯0.3011
    4              13         12      2.5           0.877      ¯0.3506
    5              12         11      2.25          0.7893     ¯0.6006
    6              11         10      2.25          0.7893     ¯0.6006
    7               5         14      2.75          0.9647     ¯0.1006
    8              10          9      2.75          0.9647     ¯0.1006
    9               9          8      2.25          0.7893     ¯0.6006
   10               8          7      2.25          0.7893     ¯0.6006
   11               7          6      2.25          0.7893     ¯0.6006
   12               6          2      2.75          0.9647     ¯0.1006
   13               9         18      2.761         0.9687     ¯0.08925
   14              18         17      2.25          0.7893     ¯0.6006
   15              18         19      2.25          0.7893     ¯0.6006
   16              17         16      2.25          0.7893     ¯0.6006
   17              16         15      2.25          0.7893     ¯0.6006
   18              19         20      2.5           0.877      ¯0.3506
   19              20         21      2.25          0.7893     ¯0.6006
   20              21         22      2.25          0.7893     ¯0.6006
   21              22         23      2.25          0.7893     ¯0.6006
   22              23         24      2.25          0.7893     ¯0.6006
   23               5          1      3             1.052       0.1494
   24              18         28      3             1.052       0.1494
   25              28         27      2.25          0.7893     ¯0.6006
   26              28         29      2.25          0.7893     ¯0.6006
   27              27         26      2.25          0.7893     ¯0.6006
   28              29         30      2.25          0.7893     ¯0.6006
   29              26         25      2.25          0.7893     ¯0.6006
   30              30         31      2.25          0.7893     ¯0.6006
   31              31         32      2.25          0.7893     ¯0.6006
   32              32         33      2.25          0.7893     ¯0.6006
   33              33         34      2.25          0.7893     ¯0.6006
   34               1         37      5.891         2.067       3.04  X
   35              37         36      1.875         0.6578     ¯0.9756
   36               3         35      9.125         3.201       6.274 X
   37              35         38      9.495         3.331       6.644 X

INPUT EDGE INCONSISTENCY FACTOR (NO. STD. DEVS.)
⎕:
      1


INCONSISTENT LINKS:   34   36   37

CLUSTER NO. 1:   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15   16   17   18
       27   28   29   30   31   32   33   34

CLUSTER NO. 2:   35

CLUSTER NO. 3:   36   37

CLUSTER NO. 4:   38
```

Q93
CORY

EXTREMAL PSEUDO-OBJECT COORDINATES

POINT	X	Y
1	20	11.5
2	20	11.75
3	20	13.75
4	20	14
5	20	16
6	20	16.25
7	20	18.25
8	20	18.5
9	20	20.5
10	20	20.75
11	20	22.75
12	17.75	22.75
13	17	22.75
14	14.75	22.75
15	9.25	22.5
16	6.25	3
17	6.25	1
18	14.75	0.25
19	17	0.25
20	20	0.5
21	20	2.5
22	20	2.75
23	20	4.75
24	20	5
25	20	7
26	20	7.25
27	20	9.25
28	20	9.5
29	5	17
30	2	17
31	2	9
32	5	9
33	1.25	2.5
34	0	2.5
35	0	1
36	1.5	0
37	2	0
38	2	0.5
39	1.5	22.75
40	0.25	22.75
41	0.25	21.5
42	1.5	21.5

OBJ.	P1	PN
1	1	28
2	29	32
3	33	38
4	39	42

MOVE CARRIAGE TO TOP OF NEW PAGE. HIT CARRIAGE RETURN

46

395 CORY

46

```
        CLUSTER

TYPE IN NAME OF ROOM
395 CORY

INPUT THE ROOM COORDINATES
⎕:
          0  23  0  0  16.5  0  16.5  23

INPUT SCALE INCREMENT FACTOR
⎕:
          .25

INPUT OBJECT COORDINATES, ONE SET PER LINE.  HIT 0 AFTER LAST SET.
⎕:
         10  19  13  19  10  13  13  13
⎕:
         10  13  13  13  10  7  13  7
⎕:
         8  21  8  22.5  9.5  21  9.5  22.5
⎕:
         .5  21  .5  22.5  2  21  2  22.5
⎕:
         .5  17  .5  15.25  2.5  17  2.5  15.25
⎕:
         .5  15  .5  13.25  2.5  15  2.5  13.25
⎕:
         .5  12.75  .5  11  2.5  12.75  2.5  11
⎕:
         .5  10.75  .5  9  2.5  10.75  2.5  9
⎕:
         .5  8.75  .5  7  2.5  8.75  2.5  7
⎕:
         .5  6.75  .5  5  2.5  6.75  2.5  5
⎕:
         .5  4.75  .5  3  2.5  4.75  2.5  3
⎕:
         .5  2.75  .5  1  2.5  2.75  2.5  1
⎕:
         3  17  5  17  5  15.25  3  15.25
⎕:
         3  15  5  15  5  13.25  3  13.25
⎕:
         3  13  5  13  5  11.25  3  11.25
⎕:
         3  9.25  5  9.25  5  7.5  3  7.5
⎕:
         3  11  4.5  11  4.5  9.5  3  9.5
⎕:
         3  7.25  4.5  7.25  4.5  5.75  3  5.75
⎕:
         3  5.5  4.5  5.5  4.5  4  3  4
⎕:
         3  3.75  4.5  3.75  4.5  2.25  3  2.25
⎕:
         3  2  4.5  2  4.5  .5  3  .5
⎕:
         5  6.5  6.5  6.5  6.5  5  5  5
⎕:
         5  4.5  6.5  4.5  6.5  3  5  3
⎕:
         5  2.5  6.5  2.5  6.5  1  5  1
⎕:
         7  5.5  8.5  5.5  8.5  4  7  4

         7  3.75  8.5  3.75  8.5  2.25  7  2.25

         7  2  8.5  2  8.5  .5  7  .5

         9  2  10.5  2  10.5  .5  9  .5

         11  2  12.5  2  12.5  .5  11  .5

         9.5  18  10  18  10  16.5  9.5  16.5

         9.5  16  10  16  10  14.5  9.5  14.5

         9.5  13.5  10  13.5  10  12  9.5  12

         9.5  11.5  10  11.5  10  10  9.5  10

         9.5  9  10  9  10  7.5  9.5  7.5

         14.5  .5  16  .5  16  2  14.5  2

         0  0  1  0  1  .5  0  .5

         0  17.25  .5  17.25  .5  17.75  0  17.75

         0
```

395
CORY

EXTREMAL OBJECT COORDINATES

POINT	X	Y
1	10	19
2	13	19
3	10	13
4	13	13
5	10	13
6	13	13
7	10	7
8	13	7
9	8	21
10	8	22.5
11	9.5	21
12	9.5	22.5
13	0.5	21
14	0.5	22.5
15	2	21
16	2	22.5
17	0.5	17
18	0.5	15.25
19	2.5	17
20	2.5	15.25
21	0.5	15
22	0.5	13.25
23	2.5	15
24	2.5	13.25
25	0.5	12.75
26	0.5	11
27	2.5	12.75
28	2.5	11
29	0.5	10.75
30	0.5	9
31	2.5	10.75
32	2.5	9
33	0.5	8.75
34	0.5	7
35	2.5	8.75
36	2.5	7
37	0.5	6.75
38	0.5	5
39	2.5	6.75
40	2.5	5
41	0.5	4.75
42	0.5	3
43	2.5	4.75
44	2.5	3
45	0.5	2.75
46	0.5	1
47	2.5	2.75
48	2.5	1
49	3	17
50	5	17
51	5	15.25
52	3	15.25
53	3	15
54	5	15
55	5	13.25
56	3	13.25
57	3	13

395
CORY

58	5	13
59	5	11.25
60	3	11.25
61	3	9.25
62	5	9.25
63	5	7.5
64	3	7.5
65	3	11
66	4.5	11
67	4.5	9.5
68	3	9.5
69	3	7.25
70	4.5	7.25
71	4.5	5.75
72	3	5.75
73	3	5.5
74	4.5	5.5
75	4.5	4
76	3	4
77	3	3.75
78	4.5	3.75
79	4.5	2.25
80	3	2.25
81	3	2
82	4.5	2
83	4.5	0.5
84	3	0.5
85	5	6.5
86	6.5	6.5
87	6.5	5
88	5	5
89	5	4.5
90	6.5	4.5
91	6.5	3
92	5	3
93	5	2.5
94	6.5	2.5
95	6.5	1
96	5	1
97	7	5.5
98	8.5	5.5
99	8.5	4
100	7	4
101	7	3.75
102	8.5	3.75
103	8.5	2.25
104	7	2.25
105	7	2
106	8.5	2
107	8.5	0.5
108	7	0.5
109	9	2
110	10.5	2
111	10.5	0.5
112	9	0.5
113	11	2
114	12.5	2
115	12.5	0.5
116	11	0.5
117	9.5	18
118	10	18
119	10	16.5
120	9.5	16.5
121	9.5	16
122	10	16
123	10	14.5

395
CORY

124	9.5	14.5
125	9.5	13.5
126	10	13.5
127	10	12
128	9.5	12
129	9.5	11.5
130	10	11.5
131	10	10
132	9.5	10
133	9.5	9
134	10	9
135	10	7.5
136	9.5	7.5
137	14.5	0.5
138	16	0.5
139	16	2
140	14.5	2
141	0	0
142	1	0
143	1	0.5
144	0	0.5
145	0	17.25
146	0.5	17.25
147	0.5	17.75
148	0	17.75

OBJ.	*P1*	*PN*
1	1	4
2	5	8
3	9	12
4	13	16
5	17	20
6	21	24
7	25	28
8	29	32
9	33	36
10	37	40
11	41	44
12	45	48
13	49	52
14	53	56
15	57	60
16	61	64
17	65	68
18	69	72
19	73	76
20	77	80
21	81	84
22	85	88
23	89	92
24	93	96
25	97	100
26	101	104
27	105	108
28	109	112
29	113	116
30	117	120
31	121	124
32	125	128
33	129	132
34	133	136
35	137	140
36	141	144
37	145	148

395
CORY

51

```
WEIGHT OF MIN. SPANNING TREE  = 80.13
AVG. LINK LENGTH              = 2.226
S.D. OF LINKS                 =0.7064

LINK NO.        LINK(VI→VJ)          LINK WT.        WT./AVG.   DEV.
--------        -----------          --------        --------   ----

    1             5       37          1.858          0.8348    ¯0.3677
    2             5        6          2              0.8985    ¯0.226
    3             6        7          2.25           1.011      0.02403
    4             7        8          2              0.8985    ¯0.226
    5             8        9          2              0.8985    ¯0.226
    6             9       10          2              0.8985    ¯0.226
    7            10       11          2              0.8985    ¯0.226
    8            11       12          2              0.8985    ¯0.226
    9            12       36          1.908          0.8572    ¯0.3179
   10             8       17          2.281          1.025      0.05507
   11            17       15          1.892          0.8498    ¯0.3344
   12            17       16          1.892          0.8498    ¯0.3344
   13            16       18          1.892          0.8498    ¯0.3344
   14            18       19          1.75           0.7862    ¯0.476
   15            19       20          1.75           0.7862    ¯0.476
   16            20       21          1.75           0.7862    ¯0.476
   17            15       14          2              0.8985    ¯0.226
   18            14       13          2              0.8985    ¯0.226
   19            21       24          2.062          0.9261    ¯0.1644
   20            24       23          2              0.8985    ¯0.226
   21            23       22          2              0.8985    ¯0.226
   22            24       27          2.062          0.9261    ¯0.1644
   23            27       26          1.75           0.7862    ¯0.476
   24            26       25          1.75           0.7862    ¯0.476
   25            27       28          2              0.8985    ¯0.226
   26            28       29          2              0.8985    ¯0.226
   27            29       35          3.5            1.572      1.274
   28            25       34          4.031          1.811      1.805
   29            34        2          2.475          1.112      0.2489
   30             2       33          1.904          0.8553    ¯0.322
   31            33       32          2              0.8985    ¯0.226
   32            32       31          2.5            1.123      0.274
   33            31        1          1.904          0.8553    ¯0.322
   34            31       30          2              0.8985    ¯0.226
   35            37        4          4.366          1.961      2.14
   36            30        3          4.61           2.071      2.384

INPUT EDGE INCONSISTENCY FACTOR (NO. STD. DEVS.)
⎕:
      1


INCONSISTENT LINKS:   27   28   35   36

CLUSTER NO. 1:   1   2   30   31   32   33   34

CLUSTER NO. 2:   3

CLUSTER NO. 3:   4

CLUSTER NO. 4:   5   6   7   8   9   10   11   12   13   14   15   16   17   18   19   20   21
       36   37

CLUSTER NO. 5:   35
```

EXTREMAL PSEUDO-OBJECT COORDINATES

(395 CORY)

POINT	X	Y
1	13	13
2	13	19
3	10	19
4	9.5	18
5	9.5	16.5
6	9.5	16
7	9.5	14.5
8	9.5	13.5
9	9.5	12
10	9.5	11.5
11	9.5	10
12	9.5	9
13	9.5	7.5
14	10	7
15	13	7
16	9.5	22.5
17	8	22.5
18	8	21
19	9.5	21
20	2	22.5
21	0.5	22.5
22	0.5	21
23	2	21
24	5	17
25	0.5	17.75
26	0	17.75
27	0	17.25
28	0	0.5
29	0	0
30	1	0
31	12.5	0.5
32	12.5	2
33	16	2
34	14.5	2
35	14.5	0.5
36	16	0.5

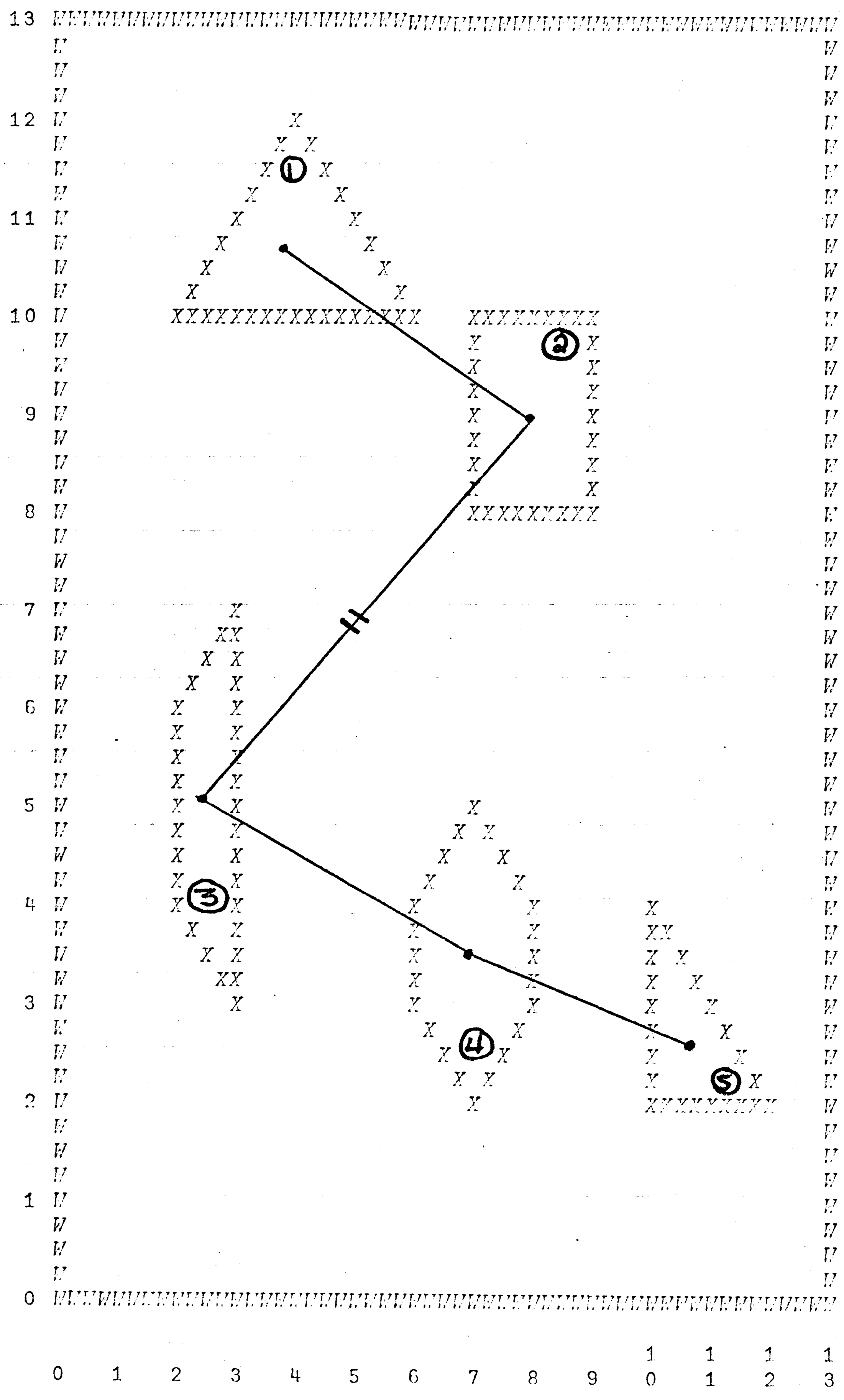
OBJ.	P1	PN
1	1	15
2	16	19
3	20	23
4	24	32
5	33	36

MOVE CARRIAGE TO TOP OF NEW PAGE. HIT CARRIAGE RETURN

```
        CLUSTER

TYPE IN NAME OF ROOM
VARIED-OBJECT ROOM

INPUT THE ROOM COORDINATES
⎕:
        0  0  0  13  13  0  13  13

INPUT SCALE INCREMENT FACTOR
⎕:
        .25

INPUT OBJECT COORDINATES, ONE SET PER LINE.  HIT 0 AFTER LAST SET.
⎕:
        2  10  4  12  6  10
⎕:
        7  10  9  10  7  8  9  8
⎕:
        3  3  3  7  2  4  2  6
⎕:
        6  3  6  4  7  2  7  5  8  3  8  4
⎕:
        10  2  10  4  12  2
⎕:
        0

MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN
```

```
                    EXTERNAL OBJECT COORDINATES
                    ---------------------------

    POINT          X          Y
    -----          -          -

      1     2   10
      2     4   12
      3     6   10
      4     7   10
      5     9   10
      6     7    8
      7     9    8
      8     3    3
      9     3    7
     10     2    4
     11     2    6
     12     6    3
     13     6    4
     14     7    2
     15     7    5
     16     8    3
     17     8    4
     18    10    2
     19    10    4
     20    12    2

  OBJ.    P1   PN
  ----    --   --

      1     1    3
      2     4    7
      3     8   11
      4    12   17
      5    18   20

  MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN
```

VARIED-OBJECT ROOM - UNCLUSTERED

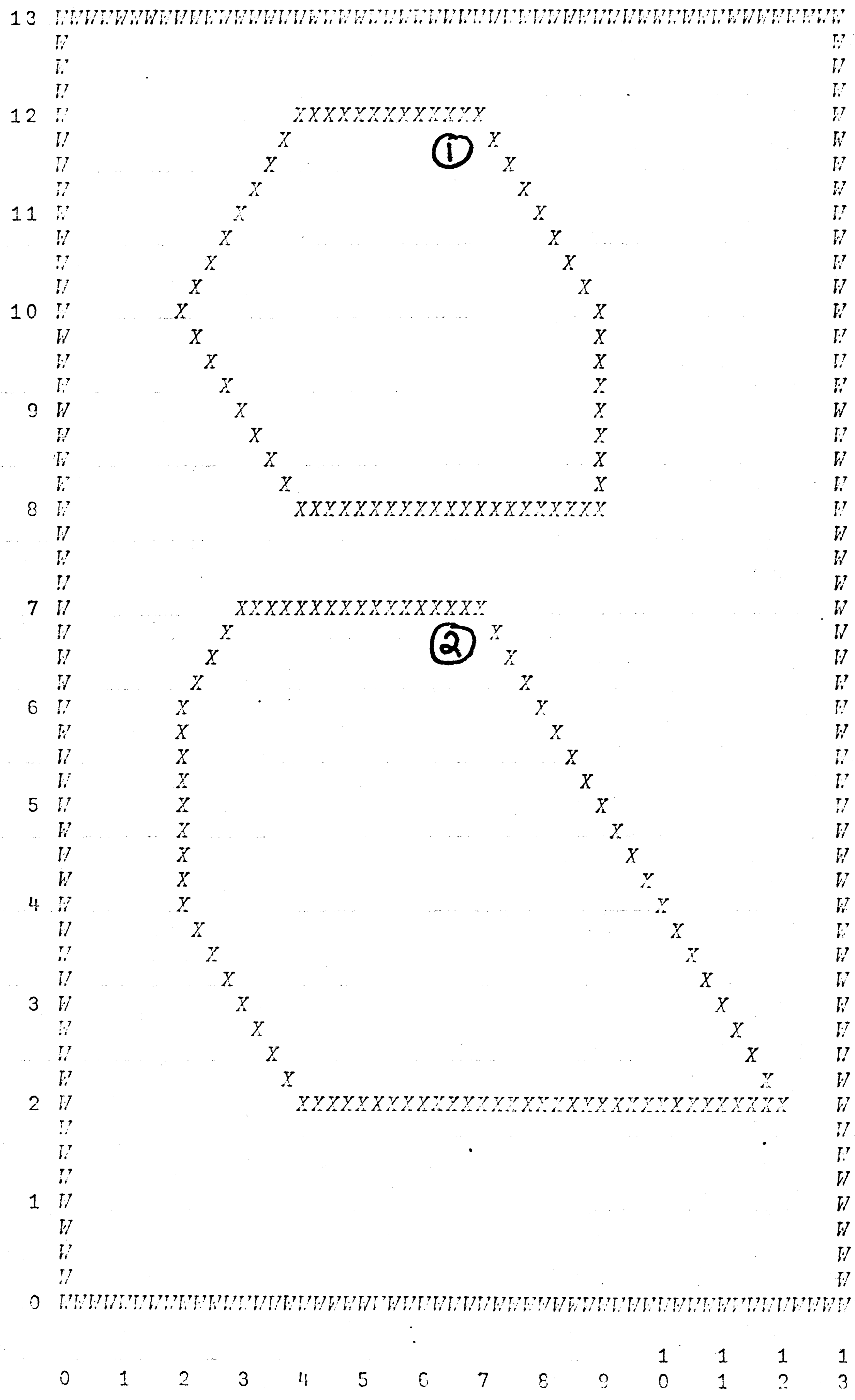


MOVE CARRIAGE TO TOP OF NEW PAGE

```
WEIGHT OF MIN. SPANNING TREE = 18.43
AVG. LINK LENGTH             = 4.607
S.D. OF LINKS                =0.8066

LINK NO.          LINK(VI→VJ)           LINK WT.        WT./AVG.    DEV.
--------          -----------           --------        --------    ----


    1               4          5           3.76          0.8162     ¯0.8466
    2               4          3           4.743         1.03        0.1366
    3               4          2           5.59          1.213       0.9834
    4               2          1           4.333         0.9406     ¯0.2734

INPUT EDGE INCONSISTENCY FACTOR (NO. STD. DEVS.)
⎕:
      1


INCONSISTENT LINKS:  3

CLUSTER NO. 1:  1  2

CLUSTER NO. 2:  3  4  5


MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN
```

```
                         EXTREMAL PSEUDO-OBJECT COORDINATES
                         ----------------------------------

       POINT        X          Y
       -----        -          -

        1    9   10
        2    4   12
        3    2   10
        4    7    8
        5    9    8
        6    3    7
        7    2    6
        8    2    4
        9    3    3
       10    7    2
       11   10    2
       12   12    2
       13   10    4


     OBJ.    P1   PN
     ----    --   --

        1    1    5
        2    6   13

     MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN
```

VARIED-OBJECT ROOM CLUSTERED



VARIED OBJECT ROOM
(PASS #2)

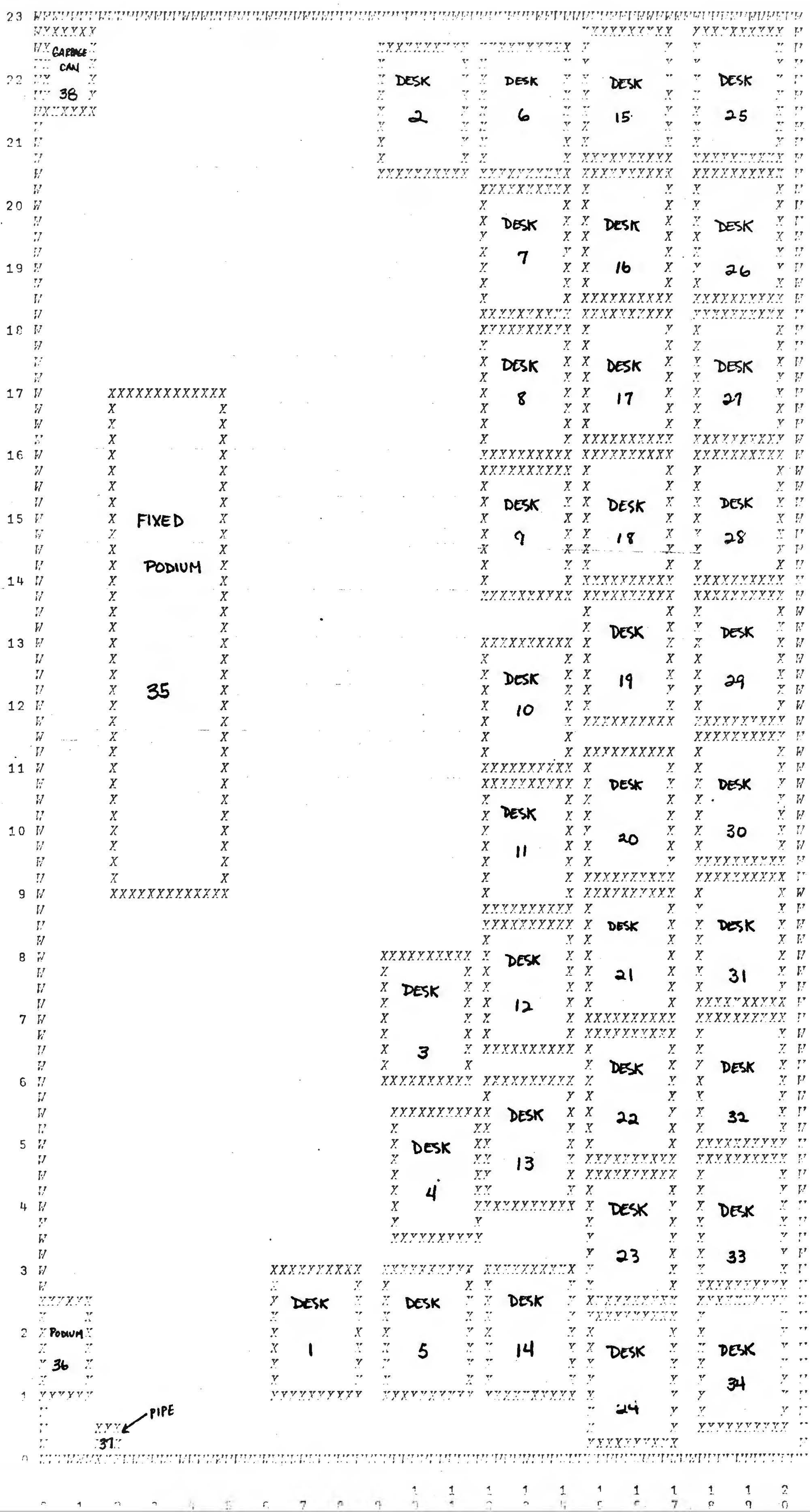
```
        CLUSTER

    TYPE IN NAME OF ROOM
    VARIED-OBJECT ROOM (PASS NO. 2)

    INPUT THE ROOM COORDINATES
    ⎕:
          0  0  0  13  13  0  13  13

    INPUT SCALE INCREMENT FACTOR
    ⎕:
          .25

    INPUT OBJECT COORDINATES, ONE SET PER LINE.  HIT 0 AFTER LAST SET.
    ⎕:
        3  3  3  7  2  4  2  6
  ⎕:
        6  3  6  4  7  2  7  5  8  3  8  4
  ⎕:
        10  2  10  4  12  2
  ⎕:
        0

        MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN
```

VARIED OBJECT ROOM
(2ND PASS)

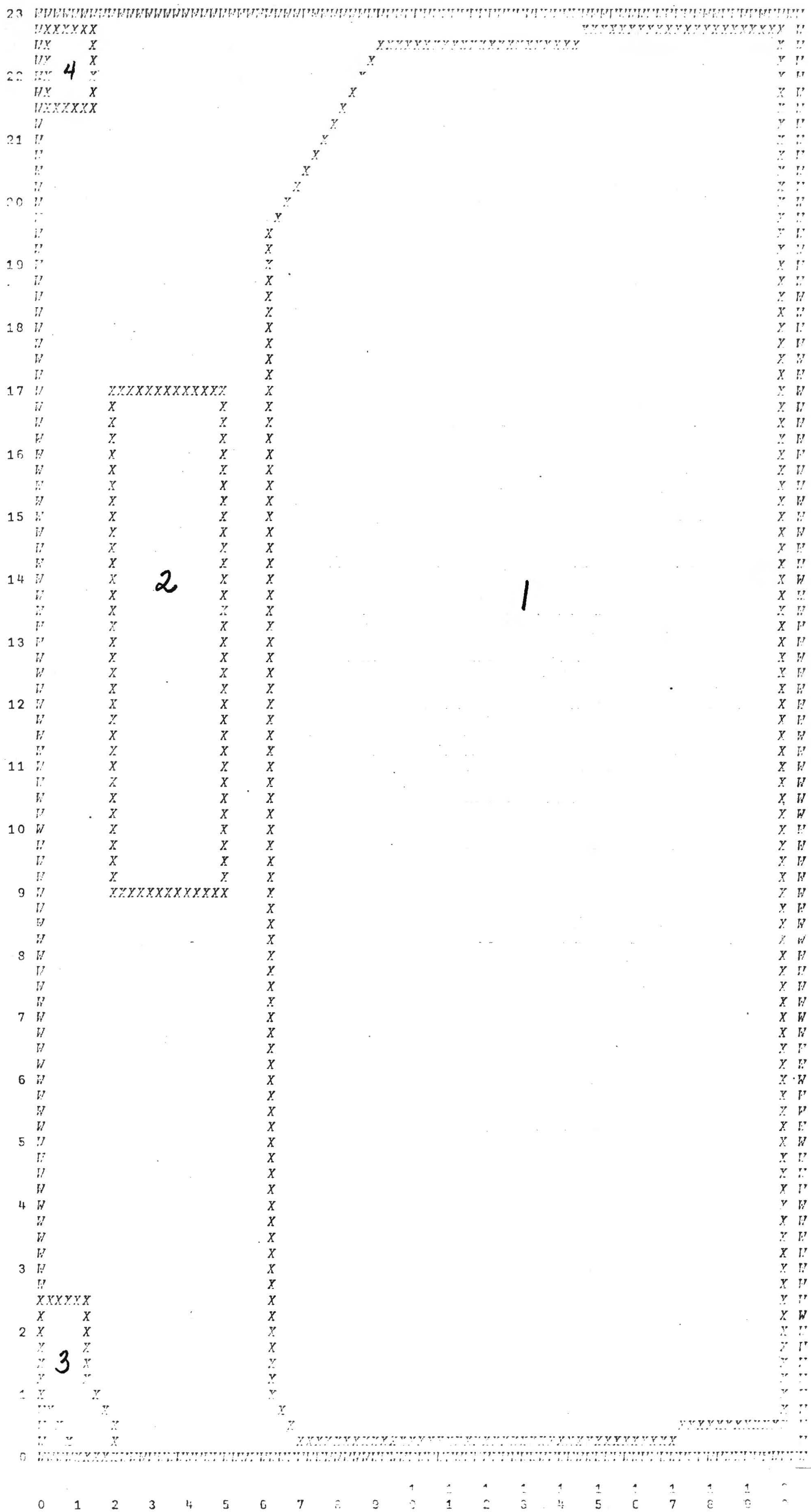
```
                EXTREMAL OBJECT COORDINATES
                ---------------------------

     POINT          X          Y
     -----          -          -

       1    3    3
       2    3    7
       3    2    4
       4    2    6
       5    6    3
       6    6    4
       7    7    2
       8    7    5
       9    8    3
      10    8    4
      11   10    2
      12   10    4
      13   12    2

     OBJ.    P1   PN
     ----    --   --

       1    1    4
       2    5   10
       3   11   13

MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN
```

```
WEIGHT OF MIN. SPANNING TREE = 8.504
AVG. LINK LENGTH             = 4.252
S.D. OF LINKS                =0.4916

LINK NO.        LINK(VI→VJ)           LINK WT.        WT./AVG.    DEV.
--------        -----------           --------        --------    ----


   1             1          2            4.743          1.116      0.4916
   2             2          3            3.76           0.8844    ¯0.4916

INPUT EDGE INCONSISTENCY FACTOR (NO. STD. DEVS.)
⎕:
      1


INCONSISTENT LINKS:  1

CLUSTER NO. 1:  1

CLUSTER NO. 2:  2  3


MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN
```

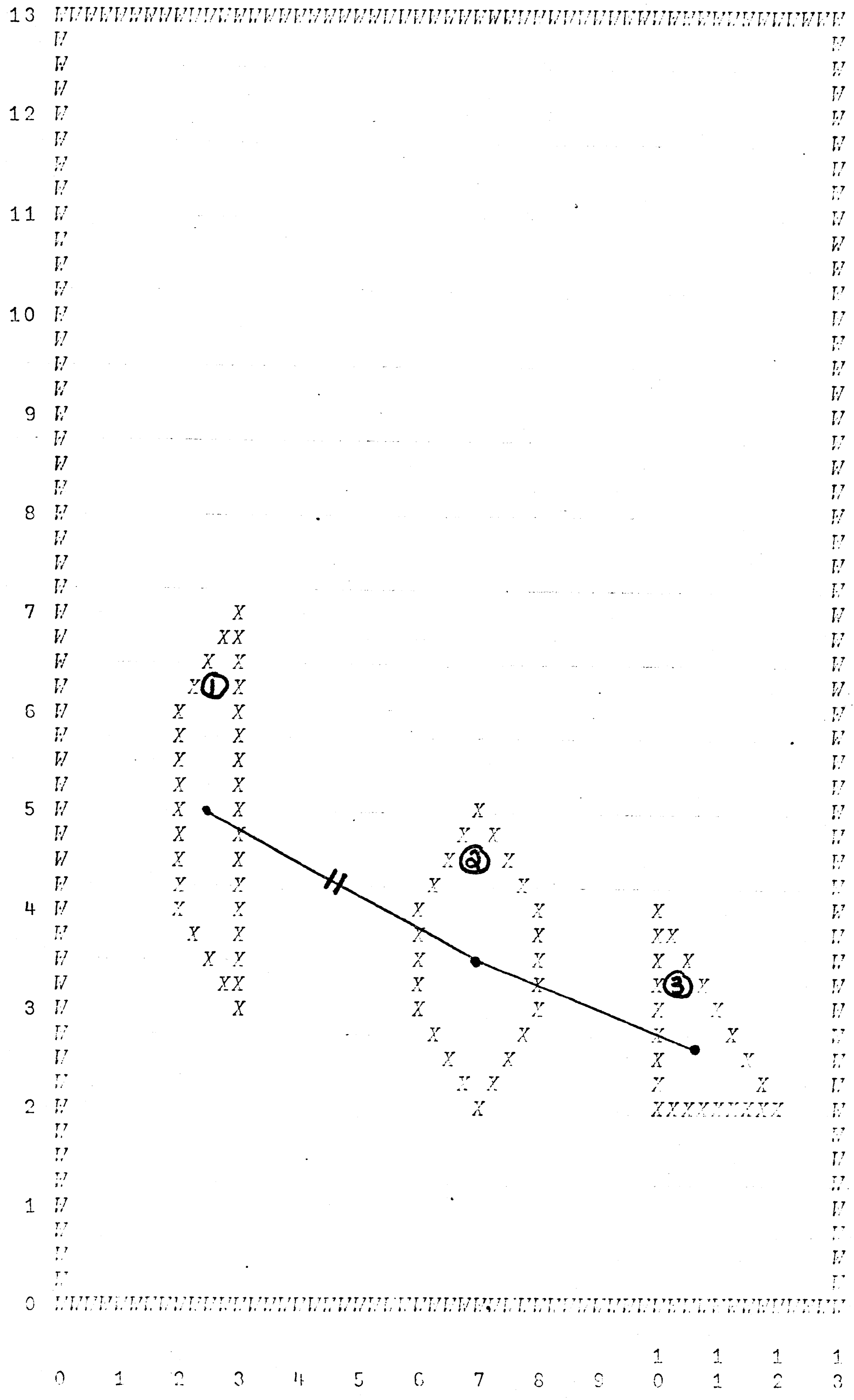
293 CORY - UNCLUSTERED



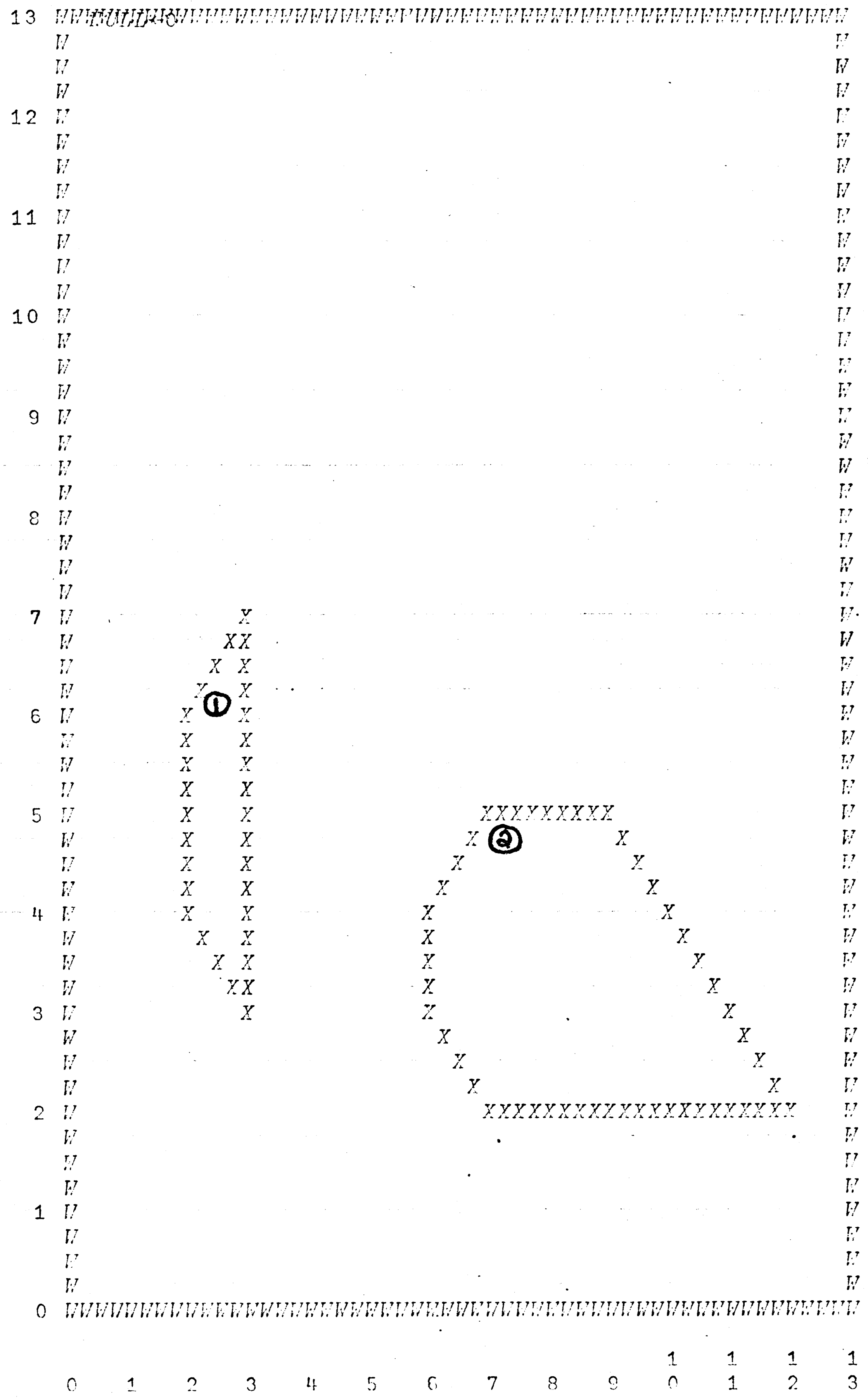
233 COPY CLUSTERED



VARIED-OBJECT ROOM (PASS NO. 2) - UNCLUSTERED



VARIED-OBJECT ROOM (PASS NO. 2) CLUSTERED



VARIED OBJECT ROOM

(PASS #2)

```
                          EXTREMAL PSEUDO-OBJECT COORDINATES
                          ----------------------------------

     POINT          X          Y
     -----          -          -

       1    3    7
       2    2    6
       3    2    4
       4    3    3
       5   10    4
       6    7    5
       7    6    4
       8    6    3
       9    7    2
      10   10    2
      11   12    2

   OBJ.     P1   PN
   ----     --   --

       1    1    4
       2    5   11

   MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN
```

```
      ∇CLUSTER[⎕]∇
    ∇ CLUSTER
[1]    NULL←0
[2]  ⍝ REQUEST NAME OF ROOM
[3]    ' '
[4]    'TYPE IN NAME OF ROOM'
[5]    ROOM←⍞
[6]  ⍝ REQUEST ROOM DIMENSION
[7]    ' '
[8]    'INPUT THE ROOM COORDINATES'
[9]    ROOMCORDS← 4 2 ρ⎕
[10]   ' '
[11]   'INPUT SCALE INCREMENT FACTOR'
[12]   INCREM←⎕
[13] ⍝ COMPUTE DIMENSIONS OF SCENE MATRIX
[14]   DIM←1+4×(⌈/ROOMCORDS[;2]),(⌈/ROOMCORDS[;1])
[15] ⍝ SET UP THE ROOM SCALES
[16]   XINDX←(4 4 ρ' '),XSCALEMAT[;ιDIM[2]]
[17]   YINDX←⌽[1](⌽[1] YSCALEMAT)[ιDIM[1];]
[18] ⍝ CREATE THE SCENE MATRIX AND PLACE THE WALL PTS
[19]   SCENEMAT←DIMρ' '
[20]   'W' CONNECT NULL ROOMCORDS
[21] ⍝ READ IN OBJECT COORDINATES, COMPUTE MIDPTS, AND PUT PTS. IN SCENE MATRIX
[22]   ' '
[23]   OBJINPUT
[24] ⍝ PRINT THE UNCLUSTERED SCENE MATRIX
[25]   ' '
[26]   'MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN'
[27]   DUM←⍞
[28]   ROOM,' - UNCLUSTERED'
[29]   (ρROOM)ρ'¯'
[30]   ' '
[31]   (YINDX,SCENEMAT),[1] XINDX
[32]   SCENEMAT←NULL
[33] ⍝ COMPUTE THE SPANNING TREE
[34]   ' '
[35]   ' '
[36]   'MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN'
[37]   DUM←⍞
[38]   SSTREE FORM MIDPTS
[39] ⍝ REQUEST THE SPANNING TREE EDGE FACTOR
[40]   ' '
[41]   'INPUT EDGE INCONSISTENCY FACTOR (NO. STD. DEVS.)'
[42]   EDGEFACT←⎕
[43] ⍝ REINITIALIZE THE SCENE MATRIX
[44]   SCENEMAT←DIMρ' '
[45]   'W' CONNECT NULL ROOMCORDS
[46] ⍝ CLUSTER THE OBJECTS
[47]   ' '
[48]   SEPARATE EDGEFACT
[49] ⍝ PRINT OUT THE PSEUDO OBJECT COORDINATES.
[50]   ' '
[51]   ' '
[52]   'MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN'
[53]   DUM←⍞
[54]   '                    EXTREMAL PSEUDO-OBJECT COORDINATES'
[55]   '                    ----------------------------------'
[56]   ' '
[57]   '   POINT          X            Y'
[58]   '   -----          -            -'
[59]   (ι(ρNCORDMAT)[1]),NCORDMAT
[60]     '
[61]     '
[
```

```
      ∇CLUSTER[⎕62]∇
[62]   'OBJ.    P1   PN'
[63]   '----    --   --'
[64]   (ι(ρNENDMAT)[1]),⌊NENDMAT
[65] ⍝ PRINT THE CLUSTERED SCENE MATRIX
[66]   ' '
[67]   ' '
[68]   'MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN'
[69]   DUM←⍞
[70]   ROOM,' - CLUSTERED'
[71]   (ρROOM)ρ'¯'
[72]   ' '
[73]   (YINDX,SCENEMAT),[1] XINDX
     ∇
```

MAIN PROGRAM

⑥

OBJECT INPUT
SUBROUTINE

```
      ∇OBJINPUT[⎕]∇
    ∇ OBJINPUT;FIRSTPT;BUFF;LASTPT;NUMPTS;NUMOBJS;OBJLABL;PTLABL
[1]     CORDMAT←BNDMAT←MIDPTS←⍳0
[2]     FIRSTPT←1
[3]     'INPUT OBJECT COORDINATES, ONE SET PER LINE.  HIT 0 AFTER LAST SET.'
[4]   INPUT:BUFF←,⎕
[5]     →(1=⍴BUFF)/MATFORM
[6]   ⍝ ASSIGN PT. NOS. TO THE OBJECT COORDINATES
[7]     CORDMAT←CORDMAT,BUFF
[8]     LASTPT←FIRSTPT+(NUMPTS←0.5×⍴BUFF)-1
[9]     BNDMAT←BNDMAT,FIRSTPT,LASTPT
[10]    FIRSTPT←LASTPT+1
[11]    BUFF←(NUMPTS,2)⍴BUFF
[12]    MIDPTS←MIDPTS,((+/BUFF[;1]),(+/BUFF[;2]))÷NUMPTS
[13]  ⍝ PUT THE OBJECT POINTS IN THE SCENE MATRIX
[14]    'X' CONNECT HULL BUFF
[15]    →INPUT
[16]  ⍝ FORM COORDINATE, MIDPT. AND OBJ. PT. MATRICES
[17]  MATFORM:CORDMAT←(LASTPT,2)⍴CORDMAT
[18]    BNDMAT←((NUMOBJS←0.5×⍴MIDPTS),2)⍴BNDMAT
[19]    MIDPTS←(NUMOBJS,2)⍴MIDPTS
[20]    ' '
[21]    OBJLABL←⍳NUMOBJS
[22]    PTLABL←⍳LASTPT
[23]  ⍝ PRINT OUT OBJECTS AND THEIR RESPECTIVE COORDINATES
[24]    ' '
[25]    'MOVE CARRIAGE TO TOP OF NEW PAGE.  HIT CARRIAGE RETURN'
[26]    DUM←⍞
[27]    ' '
[28]    '                         EXTREMAL OBJECT COORDINATES'
[29]    '                         ___________________________'
[30]    ' '
[31]    '    POINT          X            Y'
[32]    '    _____          _            _'
[33]    PTLABL,CORDMAT
[34]    ' '
[35]    'OBJ.    P1   PN'
[36]    '____    __   __'
[37]    OBJLABL,⌊BNDMAT
[38]  ⍝ GLOBAL VARIABLES: CORDMAT(NO. PTS. × 2), BNDMAT(NO. OBJS. × 2), MIDPTS(NO. OBJS. × 2)
    ∇
```

SPANNING TREE
SUBROUTINE

```
      ∇SSTREE[⎕]∇
    ∇ SSTREE CONMAT;N;MAX;SUBTREE;VINIT;LINKWTS;NNBORS;ORIGNODE;TERMNODE;MINWT;TREEWT;LINKRATS
[1]   MIDPTS←NULL
[2]   N←(⍴CONMAT)[1]
[3]   MAX←⌈/CONMAT←,CONMAT
[4]   CONMAT[CONMAT⍳0]←MAX
[5]   CONMAT←(N,N)⍴CONMAT
[6]   SPANMAT←(N,N)⍴0
[7]   SUBTREE←,VINIT←?N
[8]   CONMAT[;VINIT]←MAX
[9]   LINKWTS←LINKS←⍳0
[10] ⍝ GET NEAREST NEIGHBOR OF EACH SUBTREE NODE
[11] LOOP:NNBORS←⌊/CONMAT[SUBTREE;]
[12]  ORIGNODE←SUBTREE[NNBORS⍳MINWT←⌊/NNBORS]
[13]  TERMNODE←CONMAT[ORIGNODE;]⍳MINWT
[14]  LINKS←LINKS,ORIGNODE,TERMNODE
[15]  SPANMAT[ORIGNODE;TERMNODE]←1
[16]  LINKWTS←LINKWTS,MINWT
[17]  CONMAT[;TERMNODE]←MAX
[18]  SUBTREE←SUBTREE,TERMNODE
[19]  →((⍴LINKWTS)≠N-1)/LOOP
[20]  ' '
[21]  'WEIGHT OF MIN. SPANNING TREE = ';TREEWT←+/LINKWTS
[22]  'AVG. LINK LENGTH              = ';TREEWT←TREEWT÷N-1
[23]  'S.D. OF LINKS                 =';SDEV←((+/(LINKWTS-TREEWT)*2)÷N-1)*0.5
[24]  ' '
[25]  'LINK NO.       LINK(VI→VJ)         LINK WT.       WT./AVG.   DEV.'
      '--------       -----------         --------       --------   ----'
[26]  '
[27]  ' '
[28]  LINKS←((N-1),2)⍴LINKS
[29]  LINKDEVS←LINKWTS-TREEWT
[30]  LINKRATS←LINKWTS÷TREEWT
[31]  (⍳N-1),LINKS,⍉(3,N-1)⍴LINKWTS,LINKRATS,LINKDEVS
[32] ⍝ GLOBAL VARIABLES: SPANMAT(N × N), LINKDEVS(N-1), LINKRATS(N-1),SDEV, WHERE N=NO. OBJECTS
    ∇


      ∇FORM[⎕]∇
    ∇ CONMAT←FORM CORDMAT;X;Y
[1]   X←CORDMAT[;1]
[2]   Y←CORDMAT[;2]
[3]   CONMAT←(((X∘.-X)*2)+((Y∘.-Y)*2))*0.5
    ∇
```

CLUSTER FORMATION SUBROUTINE

```
      ∇SEPARATE[⎕]∇
    ∇ SEPARATE FACTOR;BADLINKS;BADLINK;I;J;NCLUMPS;NODES;N;ORG;TERM;CLUST;FIRSTPT;LASTPT
[1]   ⍝ DELETE INCONSISTENT EDGES
[2]   ⍝
[3]   ⍝ COMPUTE INDICES OF INCONSISTENT LINKS
[4]    BADLINKS←(LINKDEVS≥FACTOR×SDEV)/⍳⍴LINKDEVS
[5]    →((⍴BADLINKS)≠⍴LINKDEVS)/DELBAD
[6]    ' '
[7]    'NO CLUSTERING OCCURS'
[8]   ⍝ DELETE INCONSISTENT EDGES FROM SPANNING MATRIX
[9]   DELBAD:→(0=⍴BADLINKS)/TRANSCOMP
[10]   I←1
[11]  SPANDEL:BADLINK←BADLINKS[I]
[12]   SPANMAT[LINKS[BADLINK;1];LINKS[BADLINK;2]]←0
[13]   →((I←I+1)≤⍴BADLINKS)/SPANDEL
[14]   ' '
[15]   'INCONSISTENT LINKS:  ';BADLINKS
[16]   N←(⍴SPANMAT←SPANMAT∨⍉SPANMAT)[1]
[17]  ⍝ COMPUTE TRANSITIVE CLOSURE OF SPANNING MATRIX
[18]  TRANSCOMP:I←1
[19]  NEXPASS:J←1
[20]  BITCHK:→(~SPANMAT[J;I])/NEXJ
[21]   SPANMAT[J;⍳N]←SPANMAT[J;⍳N]∨SPANMAT[I;⍳N]
[22]  NEXJ:→(N≥J←J+1)/BITCHK
[23]   →(N≥I←I+1)/NEXPASS
[24]  ⍝ SPLIT THE OBJECT NODES INTO CLUSTER SETS
[25]  CLUMP:NCLUMPS←1+⍴BADLINKS
[26]   NODES←⍳N
[27]   I←1
[28]  ⍝ INITIALIZE THE NEW COORDINATE MATRIX
[29]   NCORDMAT← 1 2 ⍴0
[30]  ⍝ INITIALIZE THE OBJECT BOUNDS VECTOR
[31]   OBJPTN←⍳0
[32]  CLUSTLOOP:ORG←NODES[1]
[33]   CLUST←⍳0
[34]  SPANLOOP:CLUST←CLUST,ORG
[35]   NODES←(NODES≠ORG)/NODES
[36]   SPANMAT[;ORG]←0
[37]   TERM←SPANMAT[ORG;]⍳1
[38]   →(TERM=N+1)/CLUSTPRINT
[39]   ORG←TERM
[40]   →SPANLOOP
[41]  ⍝ PRINT OUT THE CLUSTERED NODES
[42]  CLUSTPRINT:' '
[43]   'CLUSTER NO. ';I;':  ';CLUST
[44]   ADDCLUST CLUST
[45]   →(NCLUMPS≥I←I+1)/CLUSTLOOP
[46]  ⍝ STRIP OFF NEW COORDINATE MATRIX HEADER
[47]   NCORDMAT←NCORDMAT[1+⍳(⍴NCORDMAT)[1]-1;]
[48]  ⍝ FORM MATRIX OF PSEUDO-OBJECT BOUNDS
[49]   NBNDMAT←⍳0
[50]   N←⍴OBJPTN
[51]   I←FIRSTPT←1
[52]  ADDBNDS:LASTPT←FIRSTPT+OBJPTN[I]-1
[53]   NBNDMAT←NBNDMAT,FIRSTPT,LASTPT
[54]   FIRSTPT←LASTPT+1
[55]   →(N≥I←I+1)/ADDBNDS
[56]   NBNDMAT←(N,2)⍴NBNDMAT
    ∇
```

```
      ∇ADDCLUST[⎕]∇
    ∇ ADDCLUST CLUST;OBJN;I;OBJMAT;LBND;UBND
[1]   ⍝ GET NUMBER OF OBJECTS IN CLUSTER
[2]    OBJN←⍴CLUST
[3]   ⍝ GET THE COORDINATES OF EACH OBJECT.  ADD TO MATRIX
[4]    I←1
[5]    OBJMAT← 1 2 ⍴0
[6]   OBJLOOP:LBND←BNDMAT[CLUST[I];1]
[7]    UBND←BNDMAT[CLUST[I];2]
[8]    OBJMAT←OBJMAT,[1] CORDMAT[(LBND-1)+⍳(1+UBNQ-LBND);]
[9]    →(OBJN≥I←I+1)/OBJLOOP
[10]  ⍝ STRIP OFF MATRIX HEADER
[11]   OBJMAT←OBJMAT[1+⍳(⍴OBJMAT)[1]-1;]
[12]  ⍝ COMPUTE POINTS ON CONVEX HULL
[13]   OBJMAT←HULL OBJMAT
[14]  ⍝ PLOT POINTS ON HULL PERIMETER
[15]   'X' CONNECT OBJMAT
[16]  ⍝ ADD THE NUMBER OF PTS TO THE OBJECT BOUNDS VECTOR
[17]   OBJPTN←OBJPTN,(⍴OBJMAT)[1]
[18]  ⍝ APPEND THE OBJECT COORDINATES TO THE TOTAL COORDINATE
[19]   NCORDMAT←NCORDMAT,[1] OBJMAT
    ∇
```

PSEUDO-OBJECT GENERATING SUBROUTINE

(

CONVEX HULL SUBROUTINE

```
      ∇HULL[⎕]∇
    ∇ HULLCORDS←HULL PTCORDS;X;Y;PTINDXS;RHOS;I;ANGS;PSI;ORDVEC;MARKVEC;CHKVEC;RK;RKP1;RKP2;N;XBAR;YBAR
[1]     X←PTCORDS[;1]
[2]     Y←PTCORDS[;2]
[3]     PTINDXS←⍳N←⍴X
[4]   ⍝ FIND PT. IN INTERIOR.  TRANSLATE TO (0,0)
[5]     X←X-XBAR←((⌊/X)+(⌈/X))÷2
[6]     Y←Y-YBAR←((⌊/Y)+(⌈/Y))÷2
[7]   ⍝ CALCULATE VECTOR LENGTHS
[8]     RHOS←((X×X)+(Y×Y))*0.5
[9]   ⍝ COMPUTE ANGLES FROM ○--------->
[10]    I←1
[11]    ANGS←⍳0
[12]  ANGLOOP:PSI←¯1○Y[I]÷RHOS[I]
[13]    →(X[I]≥0)/YCHK
[14]    PSI←(○1)-PSI
[15]    →NEXTANG
[16]  YCHK:→(Y[I]≥0)/NEXTANG
[17]    PSI←PSI+○2
[18]  NEXTANG:ANGS←ANGS,PSI
[19]    →(N≥I←I+1)/ANGLOOP
[20]  ⍝ PUT VECTORS IN ASCENDING ORDER BY ANGLE
[21]    ANGS←ANGS[ORDVEC←⍋ANGS]
[22]    RHOS←RHOS[ORDVEC]
[23]    X←X[ORDVEC]
[24]    Y←Y[ORDVEC]
[25]    PTINDXS←PTINDXS[ORDVEC]
[26]  ⍝ DELETE DOMINATED POINTS
[27]    ANGS←(MARKVEC←ANGS REDUCE RHOS)/ANGS
[28]    RHOS←MARKVEC/RHOS
[29]    X←MARKVEC/X
[30]    Y←MARKVEC/Y
[31]    PTINDXS←MARKVEC/PTINDXS
[32]  ⍝ SET UP MARKING VECTOR AND FIRST THREE POINTS.
[33]    CHKVEC←(N←⍴PTINDXS)⍴0
[34]    RKP2←1+RKP1←1+RK←1
[35]  ⍝ CHECK PTS. FOR INCLUSION ON HULL
[36]  TESTPT:→(PTCHK RK,RKP1,RKP2)/RKP1ON
[37]  ⍝ RK+1 NOT ON HULL.  MARK IT.
[38]    CHKVEC[RKP1]←¯1
[39]    RKP1←RK
[40]  FINDPT:RK←(RK-1)+N×(RK=1)
[41]  ⍝ IF RK IS MARKED NOT ON, GET PRECEDING POINT
[42]    →(CHKVEC[RK]≠¯1)/TESTPT
[43]    →FINDPT
[44]    →TESTPT
[45]  ⍝ RK+1 POSSIBLY ON HULL.  MARK IT.
[46]  RKP1ON:CHKVEC[RKP1]←1
[47]    RK←RKP1
[48]    RKP1←RKP2
[49]    RKP2←(RKP2+1)-N×(RKP2=N)
[50]  ⍝ IF ALL POINTS MARKED, DONE.
[51]    →(0=×/CHKVEC)/TESTPT
[52]  ⍝ FORM MATRIX OF HULL POINTS
[53]    HULLCORDS←PTCORDS[(CHKVEC=1)/PTINDXS;]
    ∇
```

SUBROUTINE FOR INTERIOR POINT ELIMINATION

```
      ∇REDUCE[⎕]∇
    ∇ MARKVEC←ANGVEC REDUCE RHOVEC;B;T;FLG
[1]     MARKVEC←(N←⍴ANGVEC)⍴1
[2]     B←1
[3]     T←2
[4]     FLG←0
[5]   ⍝ CHECK FOR MATCHING ANGLES
[6]   ANGCHK:→(ANGVEC[B]=ANGVEC[T])/RHOCHK
[7]   BMOV:B←T
[8]   TMOV:T←T+1
[9]     →(FLG=1)/0
[10]    →(T<N+1)/ANGCHK
[11]    T←FLG←1
[12]    →ANGCHK
[13]  RHOCHK:→(RHOVEC[B]>RHOVEC[T])/RTDEL
[14]  BTDEL:MARKVEC[B]←0
[15]    →BMOV
[16]  RTDEL:MARKVEC[T]←0
[17]    →TMOV
    ∇
```

```
      ∇PTCHK[⎕]∇
    ∇ ONHULL←PTCHK PTVEC;S1;S2;S3;S12;S23;SPI;SPII;ALPHA;BETA;P1;P2;P3
[1]     S1←RHOS[P1←PTVEC[1]]
[2]     S2←RHOS[P2←PTVEC[2]]
[3]     S3←RHOS[P3←PTVEC[3]]
[4]     S12←(((X[P1]-X[P2])*2)+((Y[P1]-Y[P2])*2))*0.5
[5]     S23←(((X[P2]-X[P3])*2)+((Y[P2]-Y[P3])*2))*0.5
[6]     SPI←(S1+S2+S12)÷2
[7]     SPII←(S2+S3+S23)÷2
[8]     ALPHA←2×¯2○((SPI×SPI-S1)÷S2×S12)*0.5
[9]     BETA←2×¯2○((SPII×SPII-S3)÷S2×S23)*0.5
[10]    ONHULL←(ALPHA+BETA)≤○1
    ∇
```

CONVEX HULL POINT TESTING SUBROUTINE

```
      ∇CONNECT[⎕]∇
    ∇ PLOTCHAR CONNECT HULLMAT;I;N
[1]    N←(ρHULLMAT)[1]
[2]    I←1
[3]   ADDPTS:PLOTCHAR FILL DRAW HULLMAT[I;],HULLMAT[I+1-N×(I=N);]
[4]    →(N>I←I+1)/ADDPTS
    ∇
```

PSEUDO OBJECT PLOTTING SUBROUTINE

```
      ∇FILL[⎕]∇
    ∇ FILLCHAR FILL PTMAT;N;I
[1]   ⍝ PUT POINTS IN SCENE MATRIX
[2]   ⍝
[3]   ⍝ SCALE THE POINTS
[4]    PTMAT←1+4×PTMAT
[5]    N←(ρPTMAT)[1]
[6]    I←1
[7]   LOOP:SCENEMAT[(1+DIM[1]-PTMAT[I;2]);PTMAT[I;1]]←FILLCHAR
[8]    →(N≥I←I+1)/LOOP
    ∇
```

MATRIX PICTURE GENERATING SUBROUTINE

```
      ∇DRAW[⎕]∇
    ∇ LINECORDS←DRAW PTS;BASPT;FINPT;NEWPT;DELTA
[1]    BASPT←PTS[1],PTS[2]
[2]    FINPT←PTS[3],PTS[4]
[3]    DELTA←INCREM××(FINPT-BASPT)
[4]    LINECORDS← 2 2 ρPTS
[5]    NEWPT←BASPT+DELTA
[6]   NEWCHK:→(×/NEWPT=FINPT)/0
[7]    LINECORDS←LINECORDS,[1] NEWPT
[8]    NEWPT←NEWPT+DELTA×(NEWPT≠FINPT)
[9]    →NEWCHK
    ∇
```

LINE COORDINATE GENERATING SUBROUTINE

ADDITIONAL MEASUREMENTS FOR THIRD FLOOR CORY HALL

The item number is keyed to the diagrams following the data.

1. corner to door of room 325 : 4'7"

2. door 325 : 3'6"

3. wall to room 321 : 15'11"

4. door 321 : 3'6"

5. wall to double doors : 6'5"

6. double doors : 6'8"

7. to door : 6'4"

8. bumper 4' along wall

8a. bumper 2.25" long, extends 4.5"

9. door : 5'1"

10. wall : 35'

11. recess : 1'5"

12. door : 8'

13. wall : 17'5"

14. wall : 24'6"

14a. bumper 3'1.5" from corner

15. wall 8.5"

15a. door width : 6'6"

16. corner to recess : 2'1"

18. recess 1'3"

19. corner to elevator edge : 2'2"

20. ashtray in corner : 1' by 1'

21. elevator recess : 8"

22. elevator width 42.5"

23. elevator edge to corner of large recess 2'2"

24. corner to edge of wall : 1'4"

25. wall : 19'9.5"

26. recess : 1'9"

27. door : 3'6"

28. width of hallway by elevator : 8'1"

29. wall : 12'4"

30. recess (for drinking fountain): 1'8"

30a. width of recess : 3'10.5"

31. drinking fountain , centered in recess, width : 1'2"

31a. drinking fountain protrudes 1'4"

32. wall 2'11"

33. recess : 1'9.5"

34. wall : 6'1"

35. wall : 30'1.5"

36. recess : 1'9"

37. door : 3'10"

38. wall : 41'10"

39. recess : 1'9"

40. door (room 377) : 6'9"

40a. wall : 2'4.5"

41. wall : 3'7.5"

42. hallway width : 9'10"

43. corner to doors : 5'8"

44. stairway doors : 6'4"

45. wall to corner : 1'

46. firehose extension extends 9" , center is 1'1" from corner

47. wall : 5'5"

48. wall : 5'1"

49. door 378 : 5'1"

50. wall : 3'8"

51. door 380 : 3'

52. wall : 7'4.5"

53. door 382 : 3'

54. wall : 3'8"

55. door 384 : 3'

56. wall : 7'4.5"

57. door 386 : 3'

58. wall : 3'8"

59. door 388 : 3'

60. wall : 7'4.5"

61. door 390 : 3'

62. wall : 3'8"

63. door 392 : 3'

64. wall to corner : 4'11"

65. recess to stairway (orange) : 5'4"

66. center of hose outlet is 1'2" from corner, extends 8.5"

67. wall : 1'2.5"

68. stairway doors : 6'2.5"

69. wall to corner : 5'7"

70. bumper is 3.5" long , 2.25" wide.

71. wall to bumper : 8'7"

72. bumper to bumper : 9'3"

73. wall to men's room : 12'8.5"

74. door of men's room : 3'

75. wall to bumper : 13'7.5"

76. wall to women's room : 11'9"

77. women's room door : 3'

77a. recess : 6"

78. wall : 6'9"

79. wall : 7'11"

80. wall to corner : 16'

81. hallway width : 9'10.5"

100. wall : 4'9"

101. recess : 1'10"

102. door : 6'8"

103. recess : 1'10"

104. wall : 14'2"

105. recess : 1'10"

106. door 337 : 6'

107. recess : 1'9.5"

108. wall : 15'1"

109. bookcase protrudes 1'8"

110. bookcase : 6'

110a. wall 4"

111. recess : 1'9"

112. door : 6'8"

113. recess : 1'9"

114. wall : 43'4"

115. recess : 1'9"

116. wall with fountain : 4'5.5"

117. recess : 1'9"

118. watt meter extends 1'10" and is 1'5" wide

119. water fountain extends 1'2" and is 1'4" wide

120. water fountain is 1'7.5" from room 341 wall

121. wall : 1'9"

122. recess : 1'9.5"

123. door 341 : 6'8"

124. recess : 1'9.5"

125. wall : 1'6"

126. wall : 2'2"

127. recess : 1'9.5"

128. door 353 : 6'8"

129. recess : 1'9.5"

130. display cabinet extends 1'2" from wall

131. cabinet : 9'

132. recess : 3'

133. door 355: 6'8"

134. recess : 1'9.5"

135. wall : 41'8"

136. recess : 1'10"

137. door 362 : 3'6"

138. recess : 1'10"

139. wall : 26'3"

140. recess : 1'10"

141. door 373 : 6'8"

142. recess : 1'10"

143. wall : 3'7.5"

144. hall width : 9'10"

145. wall : 17'2.5"

146. bumper 3.5" by 2"

147. bumper to bumper 9'1"

149. wall to corner : 1'8"

150. 2'9"

151. wall : 2'1"

152. recess : 10"

153. elevator : 5'6"

154. recess : 10"

155. wall : 1'9"

156. wall : 2'9"

157. wall to custodian's door : 4'4"

158. recess : 4.5"

159. door 2'10.5"

160. recess : 4.5"

161. wall to bumper : 20'

162. bumper : 3.5" by 2.5"

163. bumper to bumper : 9'2"

165. bumper to corner : 3'1"

166. wall : 5'7"

167. pink stairway recess : 4"

168. stairway door : 6'2"

169. recess : 4"

170. wall : 1'4"

171. wall : 3'4"

172. fire plug : 8" from corner

173. extends 9", 7" wide

174. wall : 3'3"

175. wall : 8"

176. recess : 2'8"

177. wall : 2'8"

178. wall : 11'3"

179. door : 3'6"

180. wall : 16'4"

181. door 338 : 2'11"

182. wall : 8'1"

183. door : 3'6"

184. wall : 7'11"

185. table 7.5" from door

185a. table width : 2'10", 6' long

186. door 332 : 3'4.5"

187. wall to bend : 7'3"

188. corner to blue stairway : 5'3"

189. door : 6'2.5"

190. recess : 5"

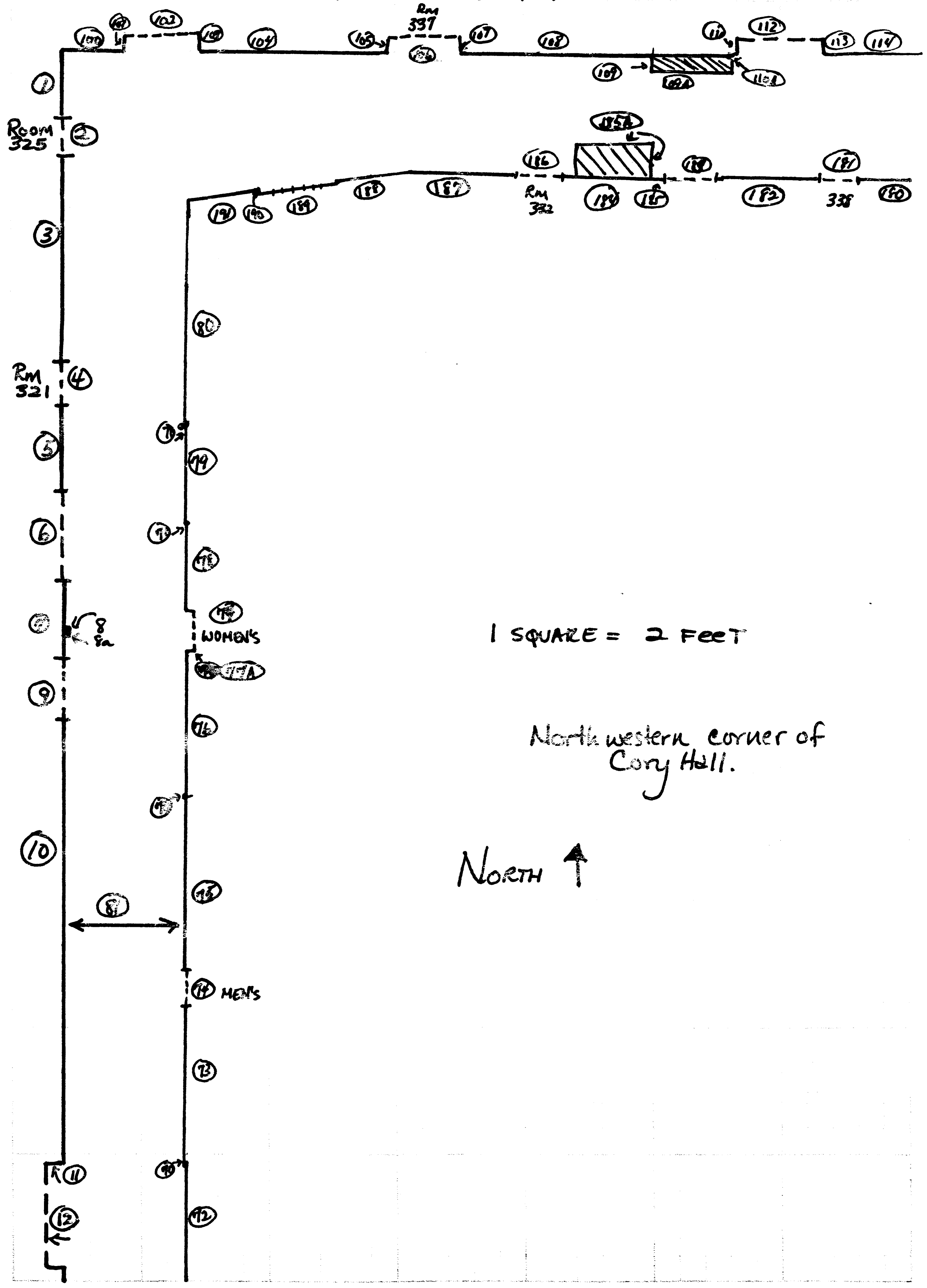
191. wall 5'6"

192. width of hall : 9'10"

193. width of hall 11'8"

How hard is it to convert to cartesian coordinates?

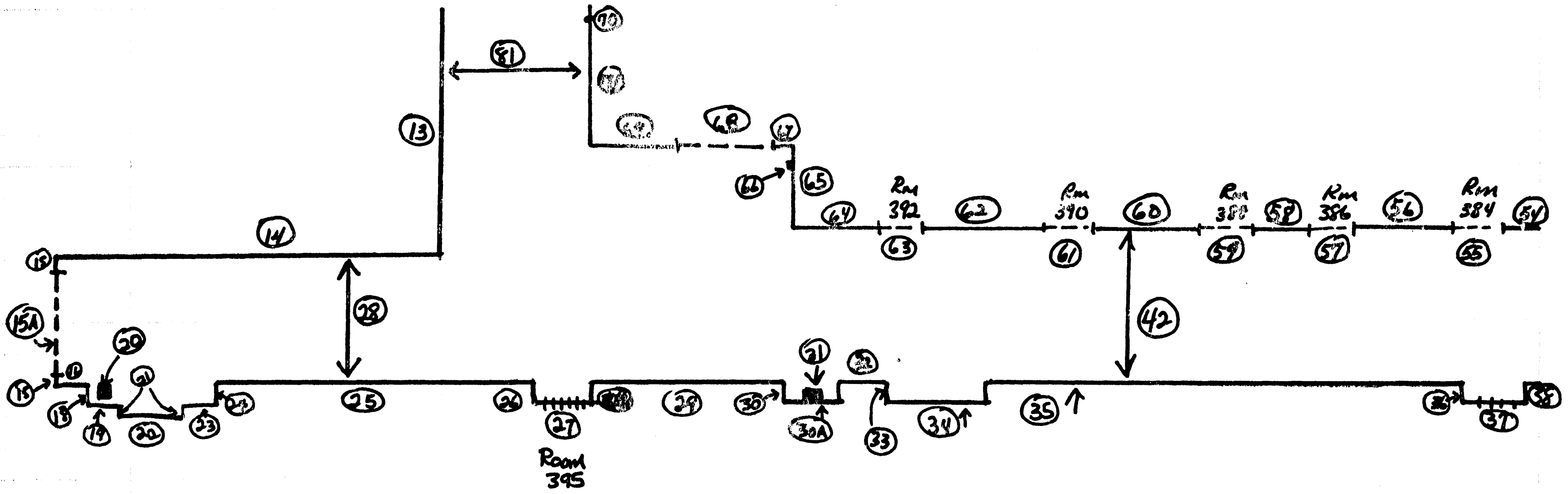
THIRD FLOOR CORY HALL



1 SQUARE = 2 FEET

Northwestern corner of Cory Hall.

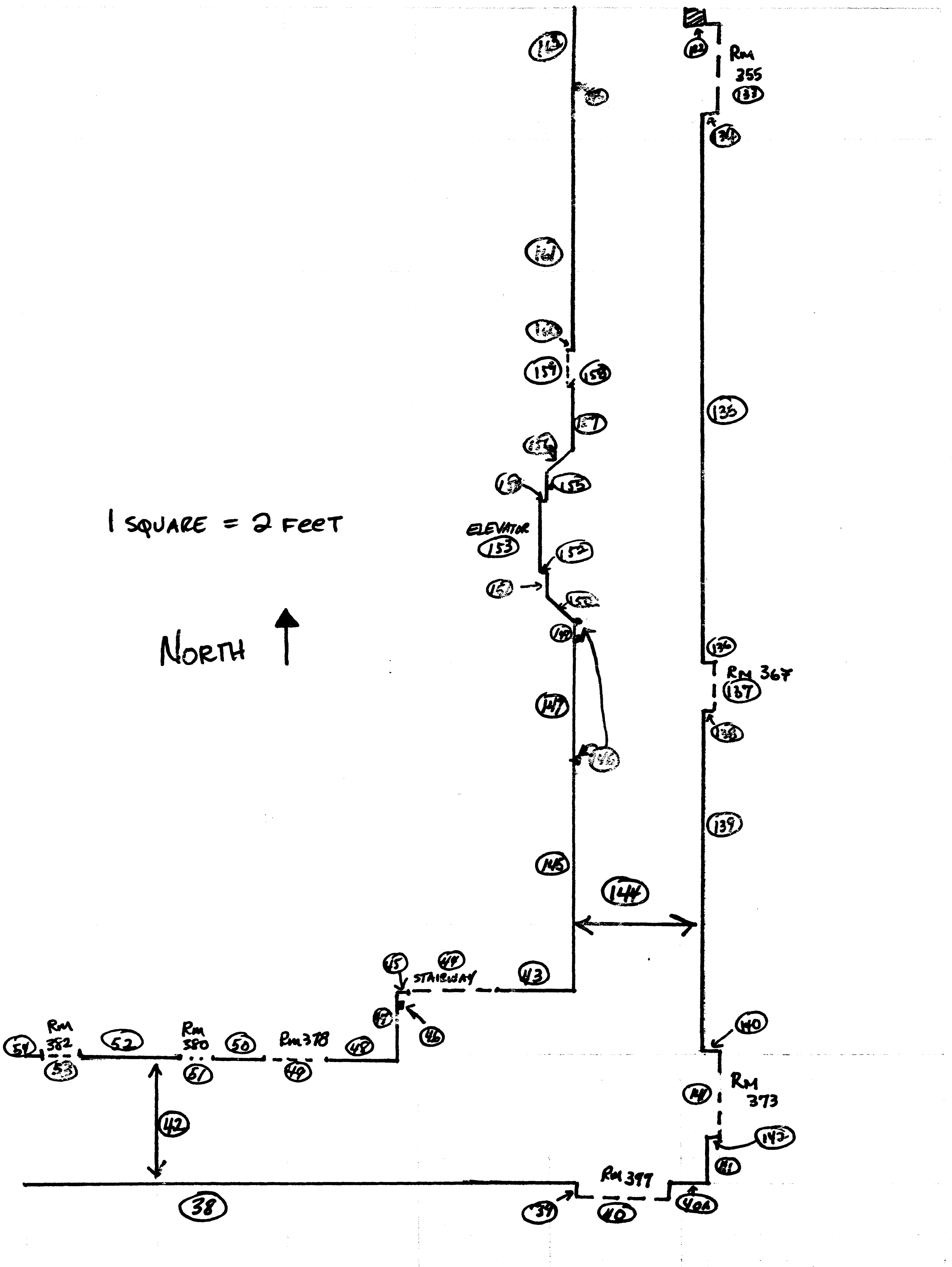
NORTH ↑



NORTH

1 SQUARE = 2 FEET

~~S~~Southwest corner of 3rd Floor Cory Hall

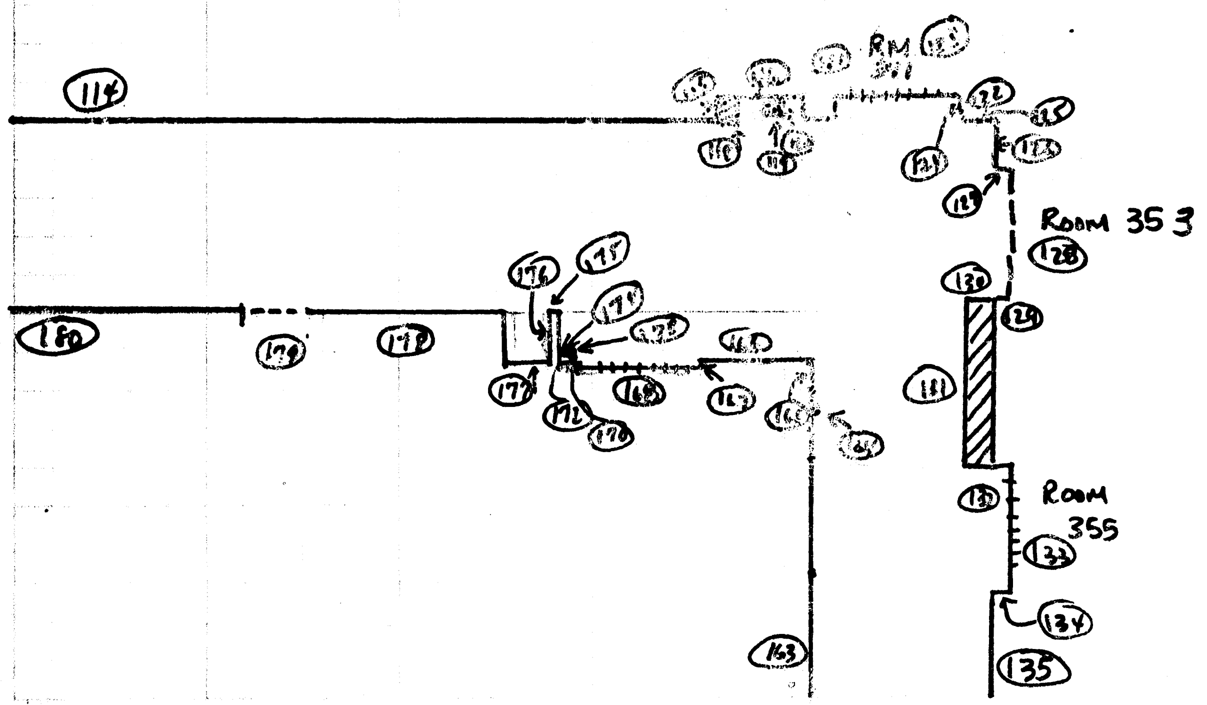


Southeast Corner of Cory Hall

Northeastern corner of Cory Hall

1 square = 2 feet

North ↑


114
RM
122
132
135
128
127
129
Room 353
128
130
129
131
132
Room 355
133
134
135
176
175
174
173
180
179
178
177
172
170
168
167
165
163