

3. Write a computer program in C, C++, Java or Python that performs hierarchical clustering for a set of objects based on their distance matrix using the single-link method. The program should implement the quad tree data structure to achieve an overall clustering time of  $O(n^2)$  for  $n$  objects  $o_1, o_2, \dots, o_n$ .

Your program should take the distance matrix input from stdin (not from a file) in the following format:

```
<d(o1,o1)>,<d(o1,o2)>,...<d(o1,on)><new line>
<d(o2,o1)>,<d(o2,o2)>,...<d(o2,on)><new line>
...
<d(on,o1)>,<d(on,o2)>,...<d(on,on)><new line>
```

Each distance value is a non-negative integer. You do not need to check for input errors. For instance, you can assume that the matrix is symmetric.

If multiple cluster pairs have the same smallest distance, break the tie **arbitrarily**. When two clusters  $C_i$  and  $C_j$  merge, with  $i < j$ , put the new distances related to the new cluster using the row/column originally occupied by  $C_i$ , and invalidate all entries originally occupied by  $C_j$  by filling in -1.

Before the first merge of clusters and after each merge, your program should print the quad tree layer by layer to the screen, each separated by a blank line. You do not need to print the object IDs.

The non-comment portion of your program is expected to contain no more than 250 lines of code.

Here is a screen shot when a sample Java program was run on an example.

```
>java QTHC
0,6,5,15,17,11,11,14,11
6,0,10,16,12,13,13,9,6
5,10,0,12,20,8,8,16,13
15,16,12,0,17,4,4,14,12
17,12,20,17,0,17,18,4,7
11,13,8,4,17,0,1,13,11
11,13,8,4,18,1,0,14,12
14,9,16,14,4,13,14,0,3
11,6,13,12,7,11,12,3,0

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

5
12,12
11,4,1
6,12,4,3

5
4,1

1

6
5,10
15,16,12
17,12,20,17
11,13,8,4,17
-1,-1,-1,-1,-1,-1
14,9,16,14,4,13,-1
11,6,13,12,7,11,-1,3

5
12,12
11,4,17
6,12,4,3

5
4,3
```

3

6

5,10

15,16,12

17,12,20,17

11,13,8,4,17

-1,-1,-1,-1,-1,-1

11,6,13,12,4,11,-1

-1,-1,-1,-1,-1,-1,-1,-1

5

12,12

11,4,17

6,12,4,-1

5

4,4

4

6

5,10

11,13,8

17,12,20,17

-1,-1,-1,-1,-1

-1,-1,-1,-1,-1,-1

11,6,13,11,4,-1,-1

-1,-1,-1,-1,-1,-1,-1,-1

5

11,8

-1,-1,-1

6,11,4,-1

5

6,4

4

6

5,10

11,13,8

11,6,13,11

-1,-1,-1,-1,-1

-1,-1,-1,-1,-1,-1

-1,-1,-1,-1,-1,-1,-1

-1,-1,-1,-1,-1,-1,-1,-1

5

6,8

-1,-1,-1

-1,-1,-1,-1

5

-1,-1

5

6

-1,-1

8,13,-1

11,6,-1,11  
-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1,-1,-1

6  
6,11  
-1,-1,-1  
-1,-1,-1,-1

6  
-1,-1

6

-1  
-1,-1  
8,-1,-1  
6,-1,-1,11  
-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1,-1,-1

-1  
6,11  
-1,-1,-1  
-1,-1,-1,-1

6  
-1,-1

6

-1  
-1,-1  
8,-1,-1  
-1,-1,-1,-1  
-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1,-1,-1

-1  
8,-1  
-1,-1,-1  
-1,-1,-1,-1

8  
-1,-1

8

-1  
-1,-1  
-1,-1,-1  
-1,-1,-1,-1  
-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1,-1  
-1,-1,-1,-1,-1,-1,-1,-1

```
-1
-1,-1
-1,-1,-1
-1,-1,-1,-1

-1
-1,-1

-1
```

Your program will be graded based on i) whether it can be compiled/run successfully, ii) whether it takes parameters as specified above, iii) its accuracy on a number of test cases and iv) whether the program is well-documented with appropriate comments. (30%)