Lecture 12
Grids

Euiseong Seo
(euiseong@skku.edu)

Grids

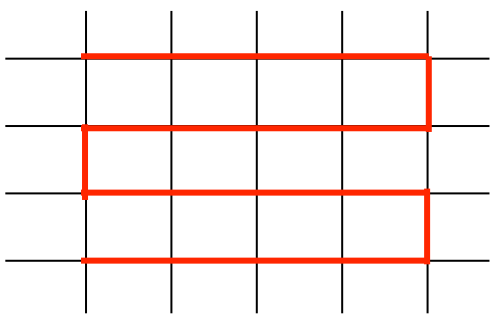
- Grids
  - Repetitive regular shapes that fill up the entire plane
  - Rectilinear, rectangular, triangular or hexagonal
- Many problems are related to grid representation
  - Map
  - VLSI layout

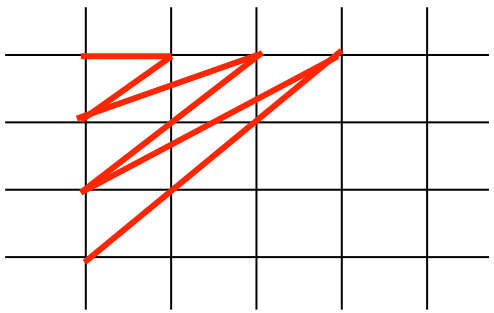
Rectilinear Grids

- Each cell is defined by horizontal and vertical lines
- Spacing between lines can be
  - Uniform
  - Non-uniform
- N-dimensional rectilinear grids
  - Each vertex touches 2N edges and 2^N cells

Representation and Traverse

- Grids are representable as a graph
  - (V, E)
  - V can be vertices or faces
  - E can be edges or neighboring relationships
- Traversal
  - Row major
  - Column major
  - Snake
  - Diagonal



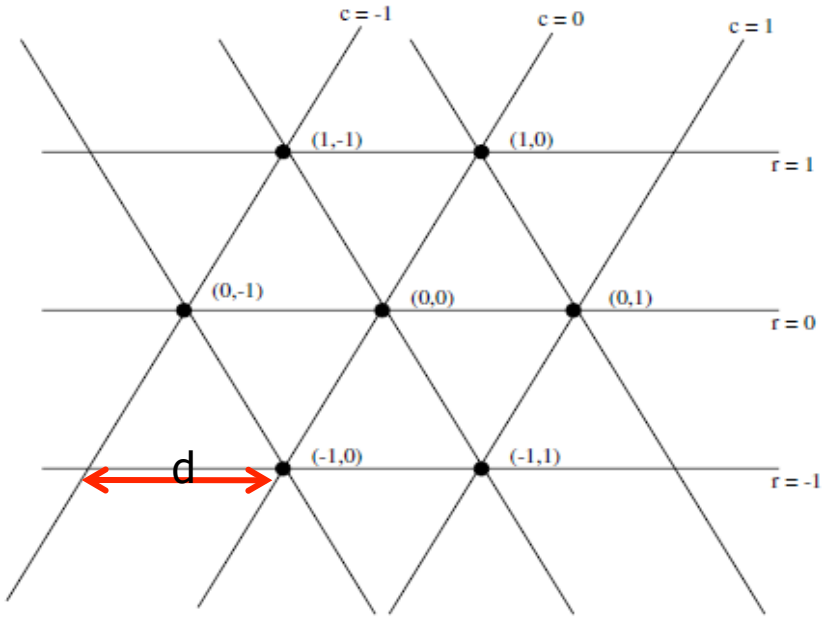


Dual Graphs

- Two-dimensional arrays are the natural choice to represent planar rectangular grids
- m[i][j] can be either
  - the (i,j)th vertex, or
  - the (i,j)th face
- m[i][j][d]
  - Adjacency representation for edge-weighted grids
  - d ranges over four values (north, east, south and west)

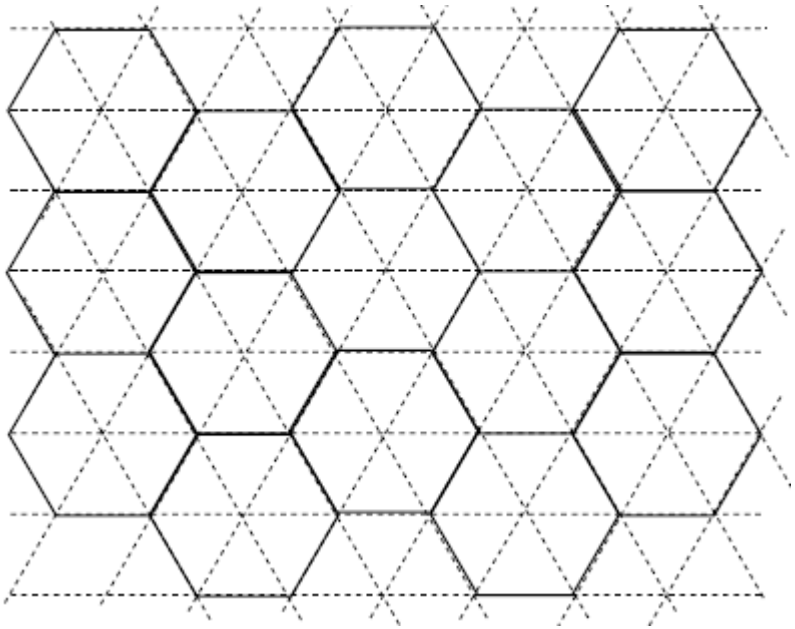
Triangular Lattices

- A vertex needs to be assigned coordinates
  - Triangular coordinates
    - (x,y)
  - Geometrical coordinates
    - $(x_g, y_g) = (d(x_t + (y_t \cos(60))), dy_t \sin(60))$



Hexagonal Lattices

- It is a subset of a triangular lattice
  - Remove every other vertex
- Interesting properties
  - It is the roundest polygon to fill the space
  - Per space/perimeter
    - A circle is the best



Hexagonal Lattices

- Representation
  - Deleting alternate vertices from a triangular lattice yields a hexagonal lattice
  - Hexagonal coordinates
  - Array coordinates
    - Conversion?

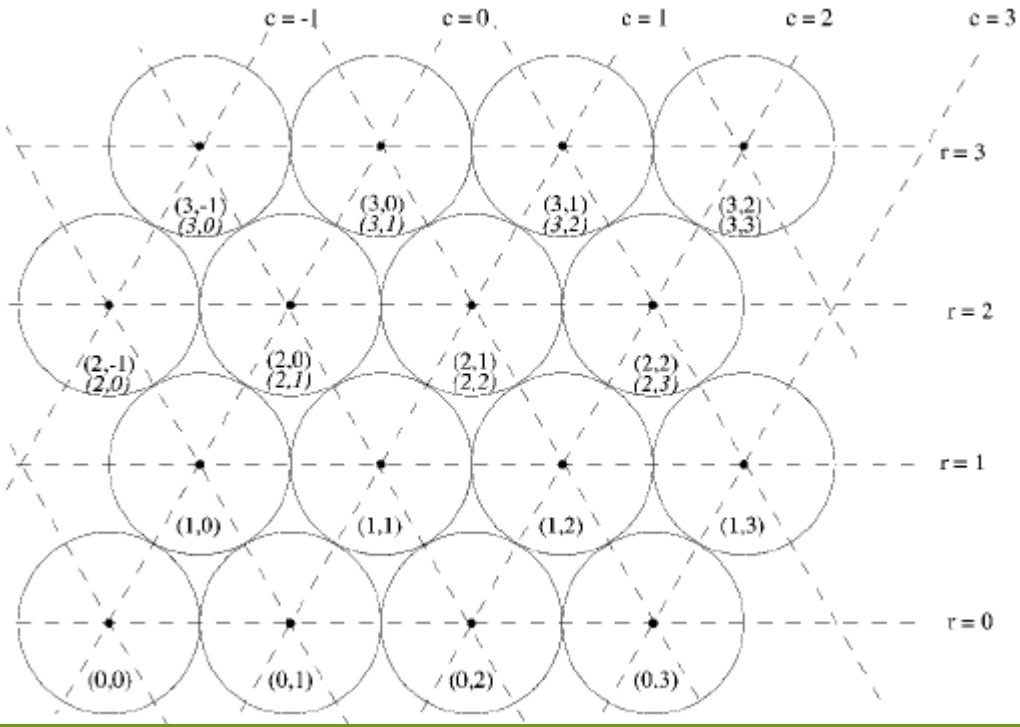
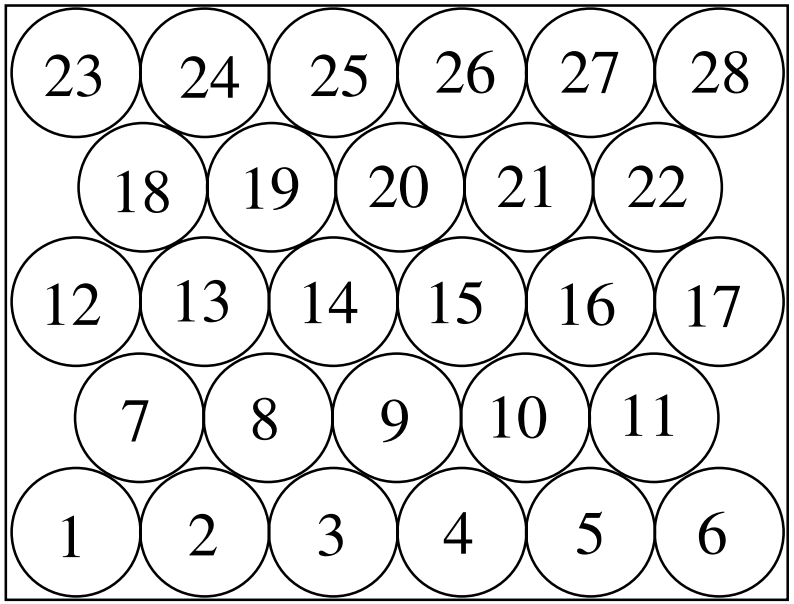


Plate Packing Problem

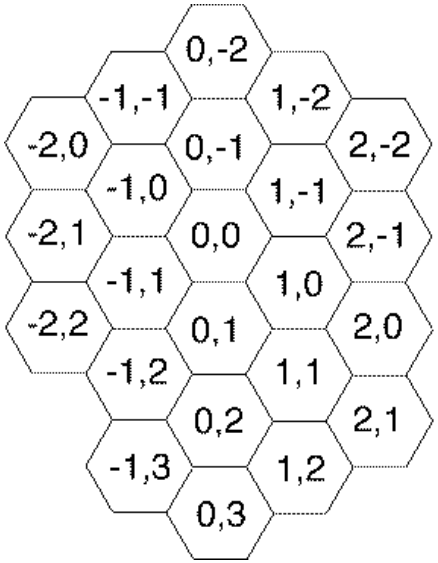
- How many plates can be packed in a *l***w* sized box?



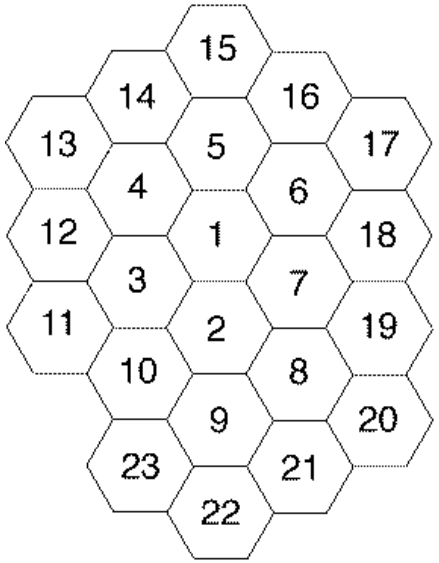
Bee Maja

Maja is a bee. She lives in a hive of hexagonal honeycombs with thousands of other bees. But Maja has a problem. Her friend Willi told her where she can meet him, but Willi (a male drone) and Maja (a female worker) have different coordinate systems:

- *Maja's Coordinate System* — Maja (on left) flies directly to a special honeycomb using an advanced two-dimensional grid over the whole hive.
- *Willi's Coordinate System* — Willi (on right) is less intelligent, and just walks around cells in clockwise order starting from 1 in the middle of the hive.



Maja's system



Willi's system

Help Maja to convert Willi's system to hers. Write a program which for a given honeycomb number returns the coordinates in Maja's system.

Bee Maja

Input

The input file contains one or more integers each standing on its own line. All honeycomb numbers are less than 100,000.

Output

Output the corresponding Maja coordinates for Willi's numbers, with each coordinate pair on a separate line.

Sample Input	*Sample Output*
1	0 0
2	0 1
3	-1 1
4	-1 0
5	0 -1

Robbery

Inspector Robostop is very angry. Last night, a bank was robbed and the robber escaped. As quickly as possible, all roads leading out of the city were blocked, making it impossible for the robber to escape. The inspector then asked everybody in the city to watch out for the robber, but the only messages he got were "We don't see him."

Robostop is determined to discover exactly how the robber escaped. He asks you to write a program which analyzes all the inspector's information to find out where the robber was at any given time.

The city in which the bank was robbed has a rectangular shape. All roads leaving the city were blocked for a certain period of time t, during which several observations of the form "The robber isn't in the rectangle R_i at time t_i" were reported. Assuming that the robber can move at most one unit per time step, try to find the exact position of the robber at each time step.

Robbery

Input

The input file describes several robberies. The first line of each description consists of three numbers W, H, and t ($1 \leq W, H, t \leq 100$), where W is the width, H the height of the city, and t is the length of time during which the city is locked.

The next line contains a single integer n ($0 \leq n \leq 100$), where n is the number of messages the inspector received. The next n lines each consist of five integers t_i, L_i, T_i, R_i, B_i, where t_i is the time at which the observation has been made ($1 \leq t_i \leq t$), and L_i, T_i, R_i, B_i are the left, top, right, and bottom, respectively, of the rectangular area which has been observed. The point $(1, 1)$ is the upper-left-hand corner, and (W, H) is the lower-right-hand corner of the city. The messages mean that the robber was not in the given rectangle at time t_i.

The input is terminated by a test case starting with $W = H = t = 0$. This case should not be processed.

Robbery

Output

For each robbery, output the line "`Robbery #`k`:`", where k is the number of the robbery. Then, there are three possibilities:

If it is impossible that the robber is still in the city, output "`The robber has escaped.`"

In all other cases, assume that the robber is still in the city. Output one line of the form "`Time step` τ`: The robber has been at` x`,`y`.`" for each time step in which the exact location can be deduced, and x and y are the column and row, respectively, of the robber in time step τ. Output these lines ordered by time τ.

If nothing can be deduced, output the line "`Nothing known.`" and hope that the inspector does not get even angrier.

Print a blank line after each processed case.

Robbery

Sample Input

```
4 4 5
4
1 1 1 4 3
1 1 1 3 4
4 1 1 3 4
4 4 2 4 4
10 10 3
1
2 1 1 10 10
0 0 0
```

Sample Output

```
Robbery #1:
Time step 1: The robber has been at 4,4.
Time step 2: The robber has been at 4,3.
Time step 3: The robber has been at 4,2.
Time step 4: The robber has been at 4,1.

Robbery #2:
The robber has escaped.
```