Programación Competitiva

Alex Josué Flórez Farfán

Magister en Ciencias de la Computación

Ciencia de la Computación – UNSA
Segundo semestre 2021

Depth First Search

10 Kinds of People

- https://open.kattis.com/problems/10kindsofpeople

```
1 4
1100
2
1 1 1 4
1 1 1 1
```

neither
decimal

```
10 20
11111111111111111111
11000000000000000101
11111111111111110000
11111111111111110000
11000000000000000111
00011111111111111111
00111111111111111111
10000000000000001111
11111111111111111111
11111111111111111111
3
2 3 8 16
8 1 7 3
1 1 10 20
```

binary
decimal
neither

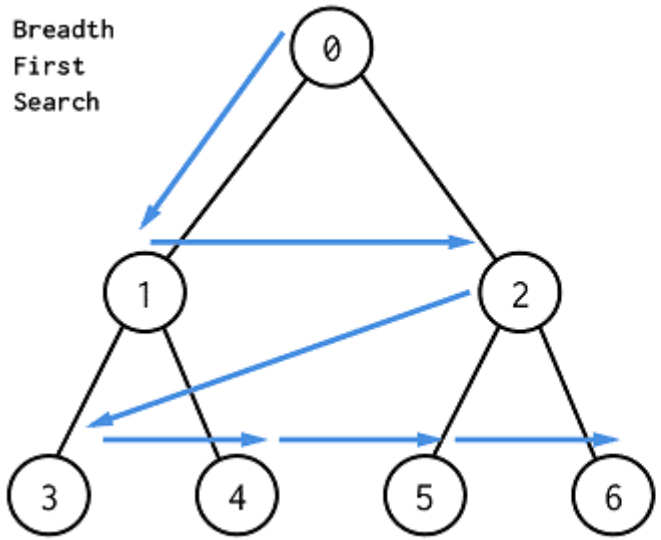
Connected Components

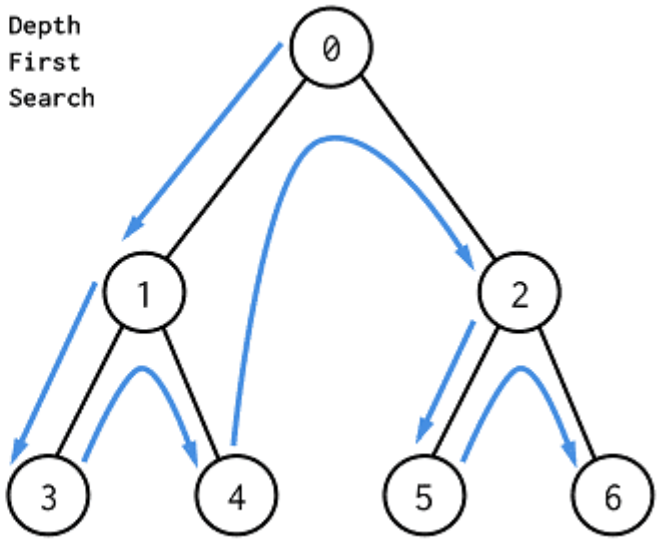
- https://www.beecrowd.com.br/judge/en/problems/view/1082

```
3

3 1
a c

10 10
a b
a c
a g
b c
c g
e d
d f
h i
i j
j h

6 4
a b
b c
c a
e f

Case #1:
a,c,
b,
2 connected components

Case #2:
a,b,c,g,
d,e,f,
h,i,j,
3 connected components

Case #3:
a,b,c,
d,
e,f,
3 connected
components
```

Breadth First Search

Breadth First Search





Breadth First Search

Breadth First search, or BFS, can be seen as the opposite of DFS.

Instead of going down a single path as in DFS,

BFS explores all possible paths at the same time.

We use a *queue* to implement a BFS.

Breadth First Search

You are on a 5 x 5 grid. If you can only move directly up/down/left/right, what is the least amount of steps it will take you to get to each cell of the grid?

-1	-1	-1	-1	-1
-1	-1	0	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

3	2	1	2	3
2	1	0	1	2
3	2	1	2	3
4	3	2	3	4
5	4	3	4	5

Breadth First Search

First, create a queue, and add the first cell to it.

Then, while there are still cells to be explored (the queue is not empty):

> Remove the top of the queue and check its four neighboring cells; if any of them have not been processed yet, then set its count to [previous cell count] + 1 and add it to the queue.

This ensures that all cells are traversed in order of 1 step away, then 2 steps away, etc.

BFS vs DFS

DFS may find a path, but it’s not guaranteed to be the *best* path.

Since BFS searches all paths at the same rate and terminates when one is found, BFS runs slower, but will find the shortest one.

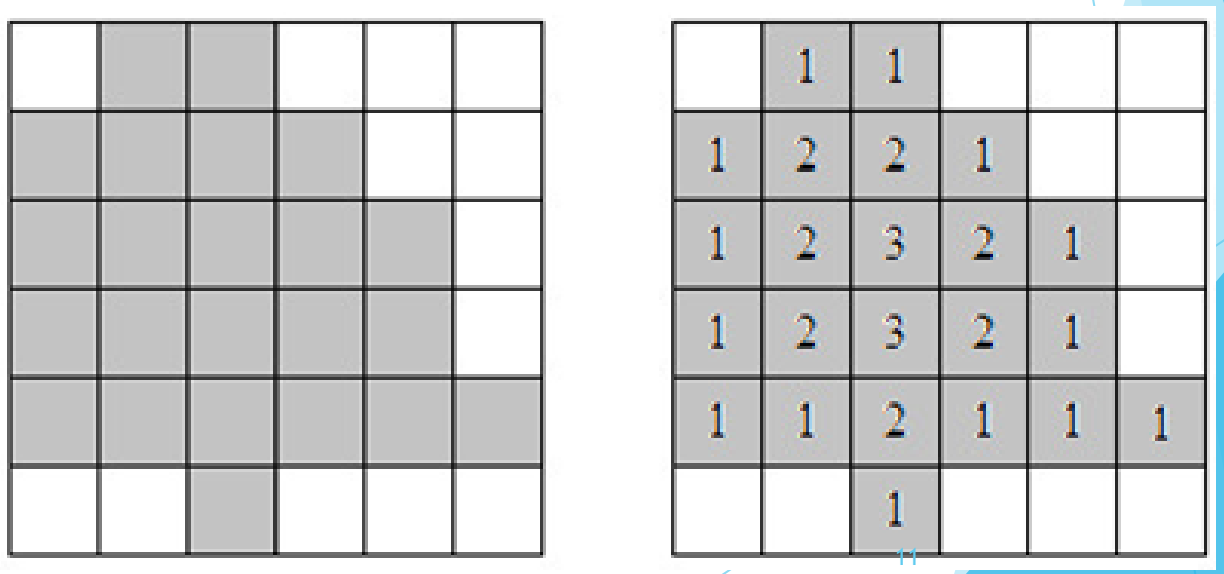
DFS can also be written recursively, but BFS cannot.

Rings

Given a cross section of a tree on a two dimensional grid, with the interior of the tree represented by a closed polygon of grid squares.

One assigns rings from the outer parts of the tree to the inner as follows: calling the non-tree grid squares “ring 0”, each ring n is made up of all those grid squares that have at least one ring (n−1) square as a neighbor (sharing a common edge).

Number of rows, columns ≤100



Rings

- First find the outer layer, which consists of a trunk cell adjacent to white square
  or, a trunk on the border
- Add cells in the outer layer to the queue

Rings

- https://open.kattis.com/problems/rings2

Grid

You are on an n×m grid where each square on the grid has a digit on it.

From a given square that has digit k on it, a move consists of jumping exactly k squares in one of the four cardinal directions.

A move cannot go beyond the edges of the grid; it does not wrap.

What is the minimum number of moves required to get from the top-left corner to the bottom-right corner?

$1 \le n,m \le 500$

2	1	2	0
1	2	0	3
3	1	1	3
1	1	2	0
1	1	1	0

Grid Using Breadth First Search

We can compute the minimal number of steps it takes to reach each square from the top-left corner.

Breadth First Search

2	1	2	0
1	2	0	3
3	1	1	3
1	1	2	0
1	1	1	0

	0	1	2	3
0	2 (0)	1	2 (1)	0
1	1 (6)	2 (4)	0 (3)	3 (5)
2	3 (1)	1 (3)	1 (2)	3 (2)
3	1 (4)	2 (4)	2 (3)	0 (5)
4	1 (5)	1 (6)	1	0 (6)

(0,0)	(0,2)	(2,0)	(2,2)	(2,3)	(2,1)	(3,2)	(1,1)	(3,1)	(3,0)
(1,3)	(4,0)	(1,0)	(4,3)						

Grid

- https://open.kattis.com/problems/grid