Day 2: Problem Analysis

Niyaz Nigmatullin

September 9, 2015

Problem A. Adjacency

Statement and solution

- Given the list of edges, find the adjacency matrix

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        a[i][j] = 0;
    }
}
for (int i = 0; i < m; i++) {
    read(u, v);
    a[u - 1][v - 1] = 1;
    a[v - 1][u - 1] = 1;
}
```

Problem B. Tree

Statement

- Given graph, check if this graph is tree

Solution is to check two of three statements:

- Graph is connected (DSU or DFS)
- Graph contains no cycle (DFS or DSU)
- $|V| = |E| + 1$

Problem C. Triangles

Statement

- Given graph, count number of triangles in graph

```
int answer = 0;
for (int i = 0; i < n; i++) {
    for (int j = i + 1; j < n; j++) {
        for (int k = j + 1; k < n; k++) {
            if (a[i][j] && a[i][k] && a[k][j]) {
                answer++;
            }
        }
    }
}
```

Problem D. Distance from the root

Statement

- Given rooted tree, count number of farthest vertices from the root

Solution

- Use DFS or BFS to find distance to every vertex
- Find maximum of distances
- Output all the vertices that has this distance

Problem E. Looking for cycle

Statement

- Given directed graph, find any cycle, if at least one exists

Solution

- Implement the algorithm described on lecture

Problem F. Finding path on a grid

Statement

- Given a grid, find the path, not necessarily shortest, from one cell to another

Solution

- Use DFS or BFS algorithm to find path
- For every v store p_v — the vertex you came to v from
- Traverse reversed path using p_v
- Don't use `std::endl`, when outputing many lines, because it flushes output every time you call it

Problem G. Knight move

Statement

- Given a $n \times m$ grid, find the number of ways for $(1, 2)$-$(2, 1)$-knight to get to the opposite corner

Solution

- Dynamic programming: $f[i][j]$ — number of ways to reach cell (i, j) from $(1, 1)$
- Formula: $f[i][j] = f[i-1][j-2] + f[i-2][j-1]$
- Output: $f[n][m]$

Problem H. Tiv tribe

Statement

- You are given strings, that represent numbers in ascending order
- The order of digits is unknown
- Numbers don't have leading zeros
- Find the order of digits, or say if it doesn't exist

Solution

- Check if numbers sorted by length
- Check if there are no equal numbers

Problem H. Tiv tribe

Solution

- Get all pairs of numbers having the same number of digits
- `acdj` and `acdf` has common prefix of 3, if `acdj` is less than `acdf`, then we know that letter `j` is less than letter `f`
- Build a graph, where letters are vertices, and there is an edge, if one letter is earlier in alphabetic order than another
- Topological sorting of this graph is an answer, handle the first step accurately, so that the first letter can be mapped to zero