Team notebook

July 10, 2024

Contents

1 Graphs **1**
- 1.1 bfs . . . 1
- 1.2 dfs . . . 1

2 Strings **1**
- 2.1 string$_h$*ashing* . . . 1

1 Graphs

1.1 bfs

Java

```
vector<vector<int>> adj; // adjacency list representation
int n; // number of nodes
int s; // source vertex

queue<int> q;
vector<bool> used(n);
vector<int> d(n), p(n);

q.push(s);
used[s] = true;
p[s] = -1;
while (!q.empty()) {
    int v = q.front();
    q.pop();
    for (int u : adj[v]) {
        if (!used[u]) {
            used[u] = true;
            q.push(u);
            d[u] = d[v] + 1;
            p[u] = v;
        }
    }
}
```

1.2 dfs

```
vector<vector<int>> adj; // graph represented as an adjacency list
int n; // number of vertices

vector<bool> visited;

void dfs(int v) {
    visited[v] = true;
    for (int u : adj[v]) {
        if (!visited[u])
            dfs(u);
    }
}
```

2 Strings

2.1 string$_h$*ashing*

```
long long compute_hash(string const& s) {
    const int p = 31;
```

```
const int m = 1e9 + 9;
long long hash_value = 0;
long long p_pow = 1;
for (char c : s) {
    hash_value = (hash_value + (c - 'a' + 1) * p_pow) % m;
    p_pow = (p_pow * p) % m;
}
return hash_value;
}
```