University of Copenhagen

string & a long

Elias Rasmussen Lolck, Thor Vejen Eriksen, William Bille Meyling

ICPC European Championship 2024

July 29, 2024

Setup

hash.sh

5246ca
```
d41d8c# hashes a file, ignoring whitespaces and comments
d41d8c# use for verifying that code is copied correctly
d41d8ccpp -dD -P -fpreprocessed | tr -d '[:space:]' | md5sum |
       cut -c-6
```

template

0acb62
```
2b74fa#include <bits/stdc++.h>
916e0dusing namespace std;
368b79typedef long long ll;
ece492#define all(x) (x).begin(), (x).end()
a1a765#define vi vector <int>
59613f#define vl vector <long long>
163402#define vvi vector <vector <int>>
29bec1#define pii pair <int, int>
66e1a7#define siz(v) (int) (v).size()
66e1a7
ce0015int main() {
9d0de8  ios::sync_with_stdio(0); cin.tie(0);
0acb62}
```

Data_structures

Disjoint Set Union

Description: Classic `DSU` using path compression and union by rank. `unite` returns true iff u and v were disjoint.
Usage: `Dsu d(n); d.unite(u, v); d.find(u);`

Complexity: `find()`, `unite()` are amortized $\mathcal{O}(\alpha(n))$, where $\alpha(n)$ is the inverse Ackermann function.

eaf77e
```
e9a6d7struct Dsu {
7c1fb1  vi p, rank;
b86822  Dsu(int n) : p(n), rank(n, 0) {
a6e3ef    iota(all(p), 0);
f17b61  }
3c38a5  int find(int x) {
cc6f17    return p[x] == x ? x : p[x] = find(p[x]);
16c2c7  }
93ff0c  bool unite(int u, int v) {
a46913    if ((u = find(u)) == (v = find(v))) return false;
4ec807    if (rank[u] < rank[v]) swap(u, v);
4a345a    rank[u] += rank[p[v] = u] == rank[v];
934347    return true;
334d56  }
eaf77e};
```