

Competitive Programming Notebook

Programadores Roblox

Contents

1	DP	2
2	String	2
3	Geometry	2
4	Graph	2
4.1	Dijkstra	2
5	Math	2
5.1	Fexp	2
6	DS	2
7	Primitives	2
8	General	2

1 DP

2 String

3 Geometry

4 Graph

4.1 Dijkstra

```

1 // Caminho mínimo com pesos positivos.
2 // Complexidade:  $O((V + E) \log V)$ .
3 vector<int> dijkstra(int S) {
4     vector<bool> vis(MAXN, 0);
5     vector<ll> dist(MAXN, LLONG_MAX);
6     dist[S] = 0;
7     priority_queue<pii, vector<pii>, greater<pii>> pq
8     ;
9     pq.push({0, S});
10    while(pq.size()) {
11        ll v = pq.top().second;
12        pq.pop();
13        if(vis[v]) continue;
14        vis[v] = 1;
15        for(auto &[peso, vizinho] : adj[v]) {
16            if(dist[vizinho] > dist[v] + peso) {
17                dist[vizinho] = dist[v] + peso;
18                pq.push({dist[vizinho], vizinho});
19            }
20        }
21        return dist;
22    }

```

```

19     }
20 }
21 return dist;
22 }

```

5 Math

5.1 Fexp

```

1 //  $a^e \bmod m$ 
2 //  $O(\log n)$ 
3
4 ll fexp(ll a, ll e, ll m) {
5     a %= m;
6     ll ans = 1;
7     while (e > 0) {
8         if (e & 1) ans = ansa % m;
9         a = aa % m;
10        e /= 2;
11    }
12    return ans%m;
13 }

```

6 DS

7 Primitives

8 General