

# Competitive Programming Notebook

Programadores Roblox

## Contents

<b>1 DP</b>	<b>2</b>
<b>2 String</b>	<b>2</b>
<b>3 Geometry</b>	<b>2</b>
<b>4 Graph</b>	<b>2</b>
4.1 Dijkstra . . . . .	2
<b>5 Math</b>	<b>2</b>
<b>6 DS</b>	<b>2</b>
<b>7 Primitives</b>	<b>2</b>
<b>8 General</b>	<b>2</b>

## 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()) {

```

```

10        ll v = pq.top().second;
11        pq.pop();
12        if(vis[v]) continue;
13        vis[v] = 1;
14        for(auto &[peso, vizinho] : adj[v]) {
15            if(dist[vizinho] > dist[v] + peso) {
16                dist[vizinho] = dist[v] + peso;
17                pq.push({dist[vizinho], vizinho});
18            }
19        }
20    }
21    return dist;
22 }

```

## 5 Math

## 6 DS

## 7 Primitives

## 8 General