Notebook - Maratona de Programação

Cabo HDMI, VGA, USB

Contents

1 Graph

1.1 BFS

Complexity: O(N + M), N: Vertices, M: Arestas
Uma bfs simples

```cpp
// TITLE: BFS
// COMPLEXITY: O(N + M), N: Vertices, M: Arestas
// DESCRIPTION: Uma bfs simples

vector<int> adj[MAX];
bool visited[MAX];

void bfs(int start)
{
    queue<int> q;
    q.push(start);

    while(not q.empty()){
        auto a = q.top();
        q.pop();
        if (visited[a]) continue;
        visited[a] = true;
        for (auto b: adj[a]){
            q.push(b);
        }
    }
}

void solve(){
// Alguma coisa
}
```

1.2 DFS

Complexity: O(N + M), N: Vertices, M: Arestas
Uma dfs simples

```cpp
// TITLE: DFS
// COMPLEXITY: O(N + M), N: Vertices, M: Arestas
// DESCRIPTION: Uma dfs simples

vector<int> adj[MAX];
void dfs(int a)
{
    if (visited[a]) return;
    visited[a] = true;

    for (auto b: adj[a]) {
        dfs(b);
    }
}

void solve(){
// Alguma coisa
}
```

2 String

2.1 Z function

Complexity: Z function complexity
z function

```cpp
// TITLE: Z function
// COMPLEXITY: Z function complexity
// DESCRIPTION: z function

void z_function(string& s)
{
    return;
}
```