C++ Handbook

Compilation of C++ Algorithms for Competitive Programming

Author:
Alan Acosta

Last Updated:
October 22, 2024

Contents

1 Template

Basic template using universal library and FASTIO.

```cpp
#include <bits/stdc++.h>
using namespace std;
#define FASTIO() ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);

void solve() {

}

int main() {
    FASTIO();
    int t = 1;
    // cin >> t;
    while (t--) solve();
}
```

2 Algebra

3 Data Structures

4 Dynamic Programming

5 String Processing

6 Linear Algebra

7 Combinatorics

8 Numerical Methods

9 Graphs

9.1 Graph traversal

9.1.1 Breadth-first search (BFS)

Search algorithm on graphs, the path found by bfs to any node is the shortest path to that node.

The algorithm works in $O(n + m)$ time, where n is number of vertices and m is the number of edges.

```cpp
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;
        }
    }
}

// For shortest path
if (!used[u]) {
    cout << "No path!";
} else {
    vector<int> path;
    for (int v = u; v != -1; v = p[v])
        path.push_back(v);
    reverse(path.begin(), path.end());
    cout << "Path: ";
    for (int v : path)
        cout << v << " ";
}
```

10 Miscellaneous