

Everyone's Connected



Contents

1	Template	2
2	Data structures	2
2.1	Simplified DSU (Stolen from GGDem)	2
2.2	Disjoint Set Union	2
2.3	Segment tree	2
2.4	Segment tree Lazy	2
2.5	Trie	2
3	Graphs	2
3.1	Graph Transversal	2
3.1.1	BFS	2
3.1.2	DFS	3
3.2	Topological Sort	3
3.3	APSP: Floyd Warshall	3
3.4	SSSP	3
3.4.1	Lazy Dijkstra	3
3.4.2	Bellman-Ford	3
3.5	Strongly Connected Components: Kosaraju	3
3.6	Articulation Points and Bridges: ModTarjan	3
4	Math	3
4.1	Identities	3

4.2	Binary Exponentiation and modArith	5
4.3	Modular Inverse (dividir mod)	5
4.4	Modular Binomial Coeficient and Permutations	5
4.5	Non-Mod Binomial Coeficient and Permutations	5
4.6	Modular Catalan Numbers	5
4.7	Ceil Fraccionario	5
4.8	Numeros de Fibonacci	5
4.9	Sieve Of Eratosthenes	5
4.10	Sieve-based Factorization	5
4.11	Cycle Finding	5
4.12	Berlekamp Massey	5
4.13	Modular Berlekamp Massey	5
4.14	Matrix exponentiation	5
4.15	Ecuaciones Diofantinas	5
4.16	Pollard-Rho, Stolen from GGDem	5
4.17	FFT, Stolen from GGDem	5
4.18	Euler Totient Function	5
5	Geometry	5
6	Strings	5
6.1	Explode by token	5
6.2	Multiple Hashings DS	5
6.3	Permute chars of string	5
6.4	Longest common subsequence	5
6.5	KMP	5
6.6	Suffix Array	5
6.7	STL Suffix Array	5
7	Classics	5
7.1	Job scheduling	5
7.1.1	One machine, linear penalty	5
7.1.2	One machine, deadlines	5
7.1.3	One machine, profit	5
7.1.4	Two machines, min time	5
8	Flow	5
8.1	Dinic, thx GGDem	5
9	Miscellaneous	5
9.1	pbds	5
9.2	Bit Manipulation	5
10	Testing	5

10.1	Gen and AutoRun testcases	5
10.1.1	Gen.cpp	5
10.1.2	Stress testing	5
10.1.3	Autorun	5
10.2	Highly Composite Numbers	5

1
 Template

```

1  #include <bits/stdc++.h>
2  #define endl '\n'
3  #define ll long long int
4  #define ull unsigned long long int
5
6  using namespace std;
7
8  /*-----SOLBEGIN-----*/
9
10 void solve() {
11     return;
12 }
13
14 int main() {
15     ios_base::sync_with_stdio(0);
16     cin.tie(0);
17
18     int t = 1; cin >> t;
19     while (t--) solve();
20
21     return 0;
22 }

```

2
 Data structures

2.1
 Simplified DSU (Stolen from GGDem)

2.2
 Disjoint Set Union

2.3
 Segment tree

2.4
 Segment tree Lazy

2.5
 Trie

3
 Graphs

3.1
 Graph Transversal

3.1.1
 BFS

```

1  #include <bits/stdc++.h>

```

```

2
3 using namespace std;
4
5 vector<bool> visited;
6 vector<vector<int>> adj;
7
8 void breadth_first_search(int node) {
9     // BFS requires queue data structure, starting from initial node 0
10    queue<int> q;
11    q.push(node);
12    visited[node] = true;
13    // While queue is not empty, pop the first element and push its
14    // children
15    while (!q.empty()) {
16        int v = q.front();
17        cout << v << " ";
18        q.pop();
19        // Push all children of v
20        for (int u : adj[v]) {
21            // If not visited, push and mark as visited
22            if (!visited[u]) {
23                q.push(u);
24                visited[u] = true;
25            }
26        }
27    }
28
29    int main() {
30        int nodes, edges;
31        cin >> nodes >> edges;
32        // Initialize visited and adjacency list
33        visited.assign(nodes, false);
34        adj.assign(nodes, vector<int>());
35        int u, v;
36        // Values of nodes starting from 0, given as pairs of nodes
37        for (int i = 0; i < edges; i++) {
38            cin >> u >> v;
39            adj[u].push_back(v);
40            adj[v].push_back(u); // Assuming undirected graph
41        }
42        breadth_first_search(0); // Start BFS from node 0
43    }

```

```

44     return 0;
45 }

```

3.1.2 DFS

3.2 Topological Sort

3.3 APSP: Floyd Warshall

3.4 SSSP

3.4.1 Lazy Dijkstra

3.4.2 Bellman-Ford

3.5 Strongly Connected Components: Kosaraju

3.6 Articulation Points and Bridges: ModTarjan

4 Math

4.1 Identities

Coeficientes binomiales.

$$(a + b)^n = \sum_{k=0}^n \binom{n}{k} a^{n-k} b^k$$

$$\binom{n}{k} = \binom{n}{n-k}$$

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

$$k \binom{n}{k} = n \binom{n-1}{k-1}$$

$$\sum_{k=0}^n n \binom{n}{k} = 2^n$$

$$\sum_{k=0}^n (-1)^k \binom{n}{k} = 0$$

$$\binom{n+m}{t} = \sum_{k=0}^t \binom{n}{k} \binom{m}{t-k}$$

$$\sum_{j=k}^n \binom{j}{k} = \binom{n+1}{k+1}$$

Numeros Catalan.

$$C_n = \frac{2(2n-1)}{n+1} C_{n-1}$$

$$C_n = \frac{1}{n+1} \binom{2n}{n}$$

$$C_n \sim \frac{4^n}{n^{3/2} \sqrt{\pi}}$$

$$\Sigma(n) = O(\log(\log(n))) \text{ (number of divisors of } n)$$

$$F_{2n+1} = F_n^2 + F_{n+1}^2$$

$$F_{2n} = F_{n+1}^2 - F_n^2$$

$$\sum_{i=1}^n F_i = F_{n+2} - 1$$

$$F_{n+i} F_{n+j} - F_n F_{n+i+j} = (-1)^n F_i F_j$$

(Möbius Function)

0 if n is square-free

1 if n got even amount of distinct prime factors

0 if n got odd amount of distinct prime factors

(Möbius Inv. Formula)

Let $g(n) = \sum_{d|n} f(d)$, then $f(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right)g(d)$.

Permutaciones objetos repetidos

$$P(n, k) = \frac{P(n, k)}{n_1! n_2! \dots}$$

Separadores, Ecuaciones lineales a variables $= b$

$$\binom{a}{b} = \binom{a+b-1}{b} = \binom{a+b-1}{a-1}$$

Teorema chino

sean $\{n_1, n_2, \dots, n_k\}$ primos relativos

$$P = n_1 \cdot n_2 \cdot \dots \cdot n_k$$

$$P_i = \frac{P}{n_i}$$

$$x \cong a_1(n_1)$$

$$x \cong a_2(n_2) \dots x \cong a_k(n_k)$$

$P_1 S_1 \cong 1(n_1)$ Donde S soluciones.

$$x = P_1 S_1 a_1 + P_2 S_2 a_2 \dots P_k S_k a_k$$

4.2 Binary Exponentiation and modArith

4.3 Modular Inverse (dividir mod)

4.4 Modular Binomial Coefficient and Permutations

4.5 Non-Mod Binomial Coefficient and Permutations

4.6 Modular Catalan Numbers

4.7 Ceil Fraccionario

4.8 Numeros de Fibonacci

4.9 Sieve Of Eratosthenes

4.10 Sieve-based Factorization

4.11 Cycle Finding

4.12 Berlekamp Massey

4.13 Modular Berlekamp Massey

4.14 Matrix exponentiation

4.15 Ecuaciones Diofantinas

4.16 Pollard-Rho, Stolen from GGDem

4.17 FFT, Stolen from GGDem

4.18 Euler Totient Function

5 Geometry

6 Strings

6.1 Explode by token

6.2 Multiple Hashings DS

6.3 Permute chars of string

6.4 Longest common subsequence

6.5 KMP

6.6 Suffix Array

6.7 STL Suffix Array