.

Everyone's Connected

Contents

1 Template **1**

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 . . . 2
- 3.2 Topological Sort . . . 2
- 3.3 APSP: Floyd Warshall . . . 2
- 3.4 SSSP . . . 2
  - 3.4.1 Lazy Dijkstra . . . 2
  - 3.4.2 Bellman-Ford . . . 2
- 3.5 Strongly Connected Components: Kosaraju . . . 2
- 3.6 Articulation Points and Bridges: ModTarjan . . . 2

4 Math **2**
- 4.1 Identities . . . 2
- 4.2 Binary Exponentiation and modArith . . . 3
- 4.3 Modular Inverse (dividir mod) . . . 3
- 4.4 Modular Binomial Coeficient and Permutations . . . 3
- 4.5 Non-Mod Binomial Coeficient and Permutations . . . 3
- 4.6 Modular Catalan Numbers . . . 3
- 4.7 Ceil Fraccionario . . . 3
- 4.8 Numeros de Fibonacci . . . 3
- 4.9 Sieve Of Eratosthenes . . . 3
- 4.10 Sieve-based Factorization . . . 3
- 4.11 Cycle Finding . . . 3
- 4.12 Berlekamp Massey . . . 3
- 4.13 Modular Berlekamp Massey . . . 3
- 4.14 Matrix exponentiation . . . 3
- 4.15 Ecuaciones Diofantinas . . . 3
- 4.16 Pollard-Rho, Stolen from GGDem . . . 3
- 4.17 FFT, Stolen from GGDem . . . 3
- 4.18 Euler Totient Function . . . 3

5 Geometry **3**

6 Strings **3**
- 6.1 Explode by token . . . 3
- 6.2 Multiple Hashings DS . . . 3
- 6.3 Permute chars of string . . . 3
- 6.4 Longest common subsequence . . . 3
- 6.5 KMP . . . 3
- 6.6 Suffix Array . . . 3
- 6.7 STL Suffix Array . . . 3

7 Clasicos **3**
- 7.1 Job scheduling . . . 3
  - 7.1.1 One machine, linear penalty . . . 3
  - 7.1.2 One machine, deadlines . . . 3
  - 7.1.3 One machine, profit . . . 3
  - 7.1.4 Two machines, min time . . . 3

8 Flow **3**
- 8.1 Dinic, thx GGDem . . . 3

9 Miscellaneous **3**
- 9.1 pbds . . . 3
- 9.2 Bit Manipulation . . . 3

10 Testing **3**

10.1 Gen and AutoRun testcases . . . 3
10.1.1 Gen.cpp . . . 3
10.1.2 Stress testing . . . 3
10.1.3 Autorun . . . 3
10.2 Highly Composite Numbers . . . 3

1 Template

```cpp
#include <bits/stdc++.h>
#define ll long long int
#define ull unsigned long long int

using namespace std;

void solve() {
    return;
}

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);

    int t = 1; cin >> t;
    while (t--) solve();

    return 0;
}
```

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

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.

$$(a+b)^n = \Sigma_{k=0}^{n} \binom{n}{k} a^{n-k} b^k$$
$$\binom{n}{k} = \binom{n}{n-k}$$
$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$
$$k\binom{n}{k} = n\binom{n-1}{k-1}$$
$$\Sigma_{k=0} n\binom{n}{k} = 2^n$$
$$\Sigma_{k=0}^{n} (-1)^k \binom{n}{k} = 0$$
$$\binom{n+m}{t} = \Sigma_{k=0}^{t} \binom{n}{k}\binom{m}{t-k}$$
$$\Sigma_{j=k}^{n} \binom{j}{k} = \binom{n+1}{k+1}$$

Numeros Catalanes.

$$C_n = \frac{2(2n-1)}{n+1} C_{n-1}$$
$$C_n = \frac{1}{n+1}\binom{2n}{n}$$
$$C_n \sim \frac{4^n}{n^{3/2}\sqrt{\pi}}$$

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

$$F_{2n+1} = F_n^2 + F_{n+1}^2$$
$$F_{2n} = F_{n+1}^2 - F_{n-1}^2$$
$$\sum_{i=1}^{n} F_i = F_{n+2} - 1$$
$$F_{n+i}F_{n+j} - F_n F_{n+i+j} = (-1)^n F_i F_j$$

(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 $f(n) = \sum d \mid ng(d)\mu\left(\frac{n}{d}\right))$.

Permutaciones objetos repetidos

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

Separadores, Ecuaciones lineares a variables $= b$

$$\left(\binom{a}{b}\right) = \binom{a+b-1}{b} = \binom{a+b-1}{a-1}$$

Teorema chino

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

$$P = n_1 \cdot n_2 \cdot ... \cdot n_k$$
$$P_i = \frac{P}{n_i}$$
$$x \cong a_1(n_1)$$
$$x \cong a_2(n_2) \; ... \; x \cong a_k(n_k)$$

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

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

4.2 Binary Exponentiation and modArith

4.3 Modular Inverse (dividir mod)

4.4 Modular Binomial Coeficient and Permutations

4.5 Non-Mod Binomial Coeficient 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