Manual de CP

insertar team

Índice

1. Plantilla

```
#include <bits/stdc++.h>
#define D(x) cout << #x << ": " << x << endl;
#define forn(i,n) for(int i=0; i< (int)n; i++)
#define for1(i,n) for(int i=1; i<= (int)n; i++)
#define all(v) v.begin(),v.end()

using namespace std;

typedef long long ll;

int main(){
    ios::sync_with_stdio(0);
    cin.tie(NULL); cout.tie(NULL);
    cout<< setprecision(20)<< fixed;
    #ifdef LOCAL
        freopen("input.txt", "r", stdin);
    #else
        #define endl '\n'
    #endif
    return 0;
}
```

2. Grafos

2.1. DFS

```
vector <int> g[MAXN];       // La lista de adyacencia
int color[MAXN];            // El arreglo de visitados
enum {WHITE, GRAY, BLACK}; // WHITE = 1, GRAY = 2, BLACK = 3

// Visita el nodo u y todos sus vecinos empezando por
// los mas profundos
void dfs(int u){
   color[u] = GRAY;    // Marcar el nodo como semi-visitado
   for (int i = 0; i < g[u].size(); ++i){
     int v = g[u][i];
     if (color[v] == WHITE) dfs(v);  // Visitar los vecinos
   }
   color[u] = BLACK;  // Marcar el nodo como visitado
}

// Llama la funcion dfs para los nodos 0 a n-1
void call_dfs(int n){
   for (int u = 0; u < n; ++u) color[u] = WHITE;
   for (int u = 0; u < n; ++u)
     if (color[u] == WHITE) dfs(u);
}
```

2.2. BFS

Complejidad: O(n+m) donde n es el numero de nodos y m es el numero de aristas

```
vector <int> g[MAXN]; // La lista de adyacencia
int d[MAXN];          // Distancia de la fuente a cada nodo

void bfs(int s, int n){ // s = fuente, n = numero de nodos
   for (int i = 0; i <= n; ++i) d[i] = -1;

   queue <int> q;
   q.push(s);
   d[s] = 0;
   while (q.size() > 0){
      int cur = q.front();
      q.pop();
      for (int i = 0; i < g[cur].size(); ++i){
         int next = g[cur][i];
```

```
            if (d[next] == -1){
                d[next] = d[cur] + 1;
                q.push(next);
            }
        }
    }
}
```