```c
// Template.c
#include <stdio.h>
#define oo 99999
#define maxn 2000
// get number of arguments with __NARG__
#define __NARG__(...) __ARG_N(__VA_ARGS__, 6,
    5, 4, 3, 2, 1, 0)
#define __ARG_N(_1, _2, _3, _4, _5, _6, N, ...)
    N
#define _VFUNC_(name, n) name##n
#define _VFUNC(name, n) _VFUNC_(name, n)
#define VFUNC(func, ...) _VFUNC(func,
    __NARG__(__VA_ARGS__))(__VA_ARGS__)
// READ
#define read1(a) scanf("%d", &a)
#define read2(a, b) scanf("%d%d", &a, &b)
#define read3(a, b, c) scanf("%d%d%d", &a, &b,
    &c)
#define read4(a, b, c, d) scanf("%d%d%d%d", &a,
    &b, &c, &d)
#define read(...) VFUNC(read, __VA_ARGS__)
// FOR
#define FOR3(i, a, b) for (int i = (a); i <=
    (b); i++)
#define FOR6(i, a, b, j, c, d) FOR3(i, a, b)
    FOR3(j, c, d)
#define FOR(...) VFUNC(FOR, __VA_ARGS__)
// PRINT
#define endline printf("\n")

// bulk set
#define setA2(a, n) FOR3(i, 1, n) a[i] = i;
#define setA3(a, n, v) FOR3(i, 1, n) a[i] = (v);
#define setA4(a, n, m, v) FOR6(i, 1, n, j, 1,
    m) a[i][j] = (v);
#define setA(...) VFUNC(setA, __VA_ARGS__)

// debug
#define debugV(a) fprintf(stderr, "#DEBUG: %s =
    %d\n", #a, a)
#define debugA2(a, i) fprintf(stderr, "#DEBUG:
    %s[%d] = %d\n", #a, i, a[i])
#define debugA3(a, i, j) \
    fprintf(stderr, "#DEBUG: %s[%d][%d] = %d\n",
        #a, i, j, a[i][j])
#define debugA(...) VFUNC(debugA, __VA_ARGS__)
#define debugLA2(a, n) FOR3(i, 1, n) debugA2(a,
    i)
#define debugLA3(a, n, m) FFOR(i, 1, n, j, 1,
    m) debugA3(a, i, j)
#define debugLA(...) VFUNC(debugLA, __VA_ARGS__)

//
#define SWAP(x, y, T) \
    do {            \
        T SWAP = x; \
        x = y;      \
        y = SWAP;   \
    } while (0)
#define max(a, b) ((a) < (b) ? b : a)
#define min(a, b) ((a) > (b) ? b : a)
typedef struct {
    int u, v, w;
} edge;

/** Stack
 * stack[++top] = u: push u to stack
 * u = stack[top--]: get u from stack
 * empty queue when top == -1;
 */
int stack[maxn], top = -1;

/** Queue
 * queue[++rear] = u : push u to queue
 * u = queue[++front]: get u from queue
 * empty queue when front >= rear;
 */
int queue[maxn], front = -1, rear = -1;

// General variables
int n, m, u, v, w, x, y;
int st, fi, budget; // start, finish
int a[maxn], b[maxn];
int cnt[maxn];

/**
 * par[u]: u's parrent;
 */
int g[maxn][maxn];
edge edges[maxn];
int par[maxn], visited[maxn];

// color graph
int color[maxn];

// dijkstra
int cost[maxn];

int main() {
    // Import Your Code Here
    return 0;
}
```