

Contents

**Intro** 2

    Template . . . . . 2

    Vim Configuration . . . . . 2

**Ad-Hoc** 2

    Debrujin . . . . . 2

    Josephus Problem . . . . . 2

    LIS . . . . . 2

    Subsets . . . . . 3

    Josephus Problem . . . . . 3

    Fib in Compile Time . . . . . 3

    Tips . . . . . 3

**Geometry** 3

**Graph** 3

**Math** 3

**String** 3

## Intro

## Template

```
#include <bits/stdc++.h>
using namespace std;

#define FILL(X, V) memset((X), (V), sizeof(X))
#define SIZE(V) int((V).size())
#define FOR2(c,i,j) for(int (c)=(i),_MAX=(j); (c)<_MAX; (c)++)
#define FOR(cont,max) FOR2((cont), 0, (max))
#define LOG(x) (31 - __builtin_clz(x))
#define W(x) cerr << "\033[31m" << #x << "=" << x << "\033[0m" << "\n";
#define ii pair<int, int>
#define ff first
#define ss second
#define oo 1e9
#define ep 1e-9
#define pb push_back

typedef long long ll
typedef unsigned long long ul

int main() {
    ios::sync_with_stdio(false);
}
```

## Vim Configuration

```
set number
set showmatch
set autoindent
set cindent
set shiftwidth=4
set smartindent
set smarttab
set softtabstop=4
set backspace=indent,eol,start
set visualbell
set hlsearch
set incsearch
set ruler
set undolevels=1000
syntax on
```

## Ad-Hoc

## Debrujin

```
string seq;

int pw(int b,int a){
    int ans = 1;
    while( a ){
        if(a&1) ans *= b;
        b *= b;
        a /= 2;
    }
    return ans;
}

void debruijn( int n, int k ){
    seq = "";
    char s[n];
    if( n == 1 ){
        for( int i = 0; i < k; i++ )
            seq += char('0'+i);
    } else {
        for( int i = 0; i < n-1; i++ )
            s[i] = k-1;

        int kn = pw(k,n-1);
        char nxt[kn]; memset(nxt,0,sizeof(nxt));
        kn *= k;
        for( int h = 0; h < kn; h++ ){
            int m = 0;
            for( int i = 0; i < n-1; i++ ){
                m *= k;
                m += s[(h+i)%(n-1)];
            }
            seq += char('0'+nxt[m]);
            s[h%(n-1)] = nxt[m];
            nxt[m]++;
        }
    }
}
```

## Josephus Problem

```
int f(int n, int k){ // Quantidade de pessoas e o tamanho do salto
    return (n == 1) ? 1 : (f(n-1, k) + k - 1) % n + 1;
}
```

## LIS

```

vector<int> lis(vector<int>& seq) {
    int smallest_end[seq.size()+1], prev[seq.size()];
    smallest_end[1] = 0;

    int sz = 1;
    for(int i = 1; i < seq.size(); ++i) {
        int lo = 0, hi = sz;
        while(lo < hi) {
            int mid = (lo + hi + 1)/2;
            if(seq[smallest_end[mid]] <= seq[i])
                lo = mid;
            else
                hi = mid - 1;
        }

        prev[i] = smallest_end[lo];
        if(lo == sz)
            smallest_end[++sz] = i;
        else if(seq[i] < seq[smallest_end[lo+1]])
            smallest_end[lo+1] = i;
    }

    vector<int> ret;
    for(int cur = smallest_end[sz]; sz > 0; cur = prev[cur], --sz)
        ret.push_back(seq[cur]);
    reverse(ret.begin(), ret.end());

    return ret;
}

```

## Subsets

```

for (int i=0; i < (1<<n); ++i) {
    for(int i2 = i; i2 > 0; i2 = (i2-1) & i) {
    }
}

```

## Josephus Problem

```

for (int i=0; i < (1<<n); ++i) {
    for(int i2 = i; i2 > 0; i2 = (i2-1) & i) {
    }
}

```

## Fib in Compile Time

```

template<ul N>
struct fibonacci : integral_constant<ul, (fibonacci<N-1>{} + fibonacci<N-2>{})> {};
template<> struct fibonacci<1> : integral_constant<ul,1> {};
template<> struct fibonacci<0> : integral_constant<ul,0> {};
#define F0(x) fib[x]=fibonacci<x>{}

```

## Tips

```

next_permutation(myints,myints+3)
prev_permutation(myints,myints+3)

```

```

bool is_power_of_2(int n) { return (n <= 0)? 0 : !(n & (n - 1)); }

```

```

scanf("%x"); // le como hexadecimal
scanf("%e"); // le como notacao cientifica

```

## Geometry

## Graph

## Math

## String