York University Team Notebook C++ (2019-2020)

Contents

1 Template

1.1 Template

```cpp
using namespace std;
#include <bits/stdc++.h>
#define ll long long
#define FOR(i,n) for(int (i)=0;(i)<(n);++(i))
#define PRE(i, m, n, in) for(int (i)=(m);(i)<(n);i+=in)
#define forn(i,n) for(int (i)=0;(i)<(n);++(i))
#define for1(i,n) for((int) i=1;i<=int(n);++i)
#define fore(i,l,r) for((int) i=l;i<=int(r);++i)
#define srt(v) sort(v.begin(),v.end())
#define printv(a) printa(a,0,a.size())
#define debug(x) cout<<#x" = "<<(x)<<endl
#define printa(a,L,R) for(int i=L;i<R;i++) cout<<a[i]<<(i==R-1?"\n":" ")
#define printv(a) printa(a,0,a.size())
#define print2d(a,r,c) for(int i=0;i<r;i++) for(int j=0;j<c;j++) cout<<a[i
    ][j]<<(j==c-1?"\n":" ")

typedef vector<string>vs;
typedef pair<ll,ll> ii;
typedef vector<ll> vl;
typedef vector<vl> vvl;
typedef vector<ii> vii;
typedef map<ll,ll> ml;
typedef map<ll, string>mpls;

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

        return 0;
}
```

2 Tree

2.1 Fenwick Tree

```cpp
//fenwick trie with range update and range sum query
void add(ll p, ll x){
    for(int i = p; i <= n; i += i & -i)
        sum1[i] += x, sum2[i] += x * p;
}
void range_add(ll l, ll r, ll x){
    add(l, x), add(r + 1, -x);
}
ll ask(ll p){
    ll res = 0;
    for(int i = p; i; i -= i & -i)
        res += (p + 1) * sum1[i] - sum2[i];
    return res;
}
ll range_ask(ll l, ll r){
    return ask(r) - ask(l - 1);
}
// two dimensional, single update, range query
void add(int x, int y, int z){
    int memo_y = y;
    while(x <= n){
        y = memo_y;
        while(y <= n)
            tree[x][y] += z, y += y & -y;
        x += x & -x;
    }
}
void ask(int x, int y){
    int res = 0, memo_y = y;
    while(x){
        y = memo_y;
        while(y)
            res += tree[x][y], y -= y & -y;
        x -= x & -x;
    }
}
// 2D, range update, single query
void add(int x, int y, int z){
    int memo_y = y;
    while(x <= n){
        y = memo_y;
        while(y <= n)
            tree[x][y] += z, y += y & -y;
        x += x & -x;
    }
}
void range_add(int xa, int ya, int xb, int yb, int z){
    add(xa, ya, z);
    add(xa, yb + 1, -z);
    add(xb + 1, ya, -z);
    add(xb + 1, yb + 1, z);
}
void ask(int x, int y){
    int res = 0, memo_y = y;
    while(x){
        y = memo_y;
        while(y)
            res += tree[x][y], y -= y & -y;
        x -= x & -x;
    }
}
// 2D, range update, range query
ll t1[N][N], t2[N][N], t3[N][N], t4[N][N];
void add(ll x, ll y, ll z){
    for(int X = x; X <= n; X += X & -X)
        for(int Y = y; Y <= m; Y += Y & -Y){
            t1[X][Y] += z;
            t2[X][Y] += z * x;
            t3[X][Y] += z * y;
            t4[X][Y] += z * x * y;
        }
}
void range_add(ll xa, ll ya, ll xb, ll yb, ll z){ //(xa, ya)     (xb, yb)
```

```
    add(xa, ya, z);
    add(xa, yb + 1, -z);
    add(xb + 1, ya, -z);
    add(xb + 1, yb + 1, z);
}
ll ask(ll x, ll y){
    ll res = 0;
    for(int i = x; i; i -= i & -i)
        for(int j = y; j; j -= j & -j)
            res += (x + 1) * (y + 1) * t1[i][j]
                - (y + 1) * t2[i][j]
                - (x + 1) * t3[i][j]
                + t4[i][j];
    return res;
}
ll range_ask(ll xa, ll ya, ll xb, ll yb){
    return ask(xb, yb) - ask(xb, ya - 1) - ask(xa - 1, yb) + ask(xa - 1,
         ya - 1);
}
```

3 String

3.1 KMP

```
vector<int> prefix_function(string s) {
  int n = (int)s.length();
  vector<int> pi(n);
  for (int i = 1; i < n; i++) {
    int j = pi[i - 1];
    while (j > 0 && s[i] != s[j]) j = pi[j - 1];
    if (s[i] == s[j]) j++;
    pi[i] = j;
  }
  return pi;
}
```