

Contents

1	Data structures	1
1.1	Segment tree	1
2	Matemáticas	1
2.1	Divisores	1
2.2	Criba de primos	2

1 Data structures

1.1 Segment tree

```
1  #define oper min
2  #define NEUT INF
3  struct STree { // segment tree for min over integers
4      vector<int> st;int n;
5      STree(int n): st(4*n+5,NEUT), n(n) {}
6      void init(int k, int s, int e, int *a){
7          if(s+1==e){st[k]=a[s];return;}
8          int m=(s+e)/2;
9          init(2*k,s,m,a);init(2*k+1,m,e,a);
10         st[k]=oper(st[2*k],st[2*k+1]);
11     }
12     void upd(int k, int s, int e, int p, int v){
13         if(s+1==e){st[k]=v;return;}
14         int m=(s+e)/2;
15         if(p<m)upd(2*k,s,m,p,v);
16         else upd(2*k+1,m,e,p,v);
17         st[k]=oper(st[2*k],st[2*k+1]);
18     }
19     int query(int k, int s, int e, int a, int b){
20         if(s>=b|e<=a)return NEUT;
21         if(s>=a&&e<=b)return st[k];
22         int m=(s+e)/2;
23         return oper(query(2*k,s,m,a,b),query(2*k+1,m,e,a,b));
24     }
25     void init(int *a){init(1,0,n,a);}
26     void upd(int p, int v){upd(1,0,n,p,v);}
27     int query(int a, int b){return query(1,0,n,a,b);}
28 }; // usage: STree rmq(n);rmq.init(x);rmq.upd(i,v);rmq.query(s,e);
```

2 Matemáticas

2.1 Divisores

```
1  // Divisores
2  vector<int> divisores = calcular_divisores(30);
3  for( int div:divisores ) {
4      cout << div << "\n";
5  }
```

El siguiente código sirve para implementar la función `calcular_divisores`. El algoritmo tiene complejidad de $O(\sqrt{n})$.

```
1 vector<int> calcular_divisores(int numero) {  
2     vector<int> divisores;  
3     for( int i=1; i*i <= numero; i++ ) {  
4         if( numero%i == 0 ) {  
5             divisores.push_back(i);  
6             if( i*i != numero ) {  
7                 divisores.push_back(numero/i);  
8             }  
9         }  
10    }  
11    return divisores;  
12 }
```

2.2 Criba de primos

La siguiente criba les servirá para calcular los primeros 78,499 primos (todos los primos menores o iguales a un millón). Si quieren saber si un número es primo, lo pueden consultar de manera constante checando el arreglo `es_primo`. Antes de usar los arreglos `es_primo` y `primos`, hay que llamar a la función `criba_primos`.