

Segment Tree

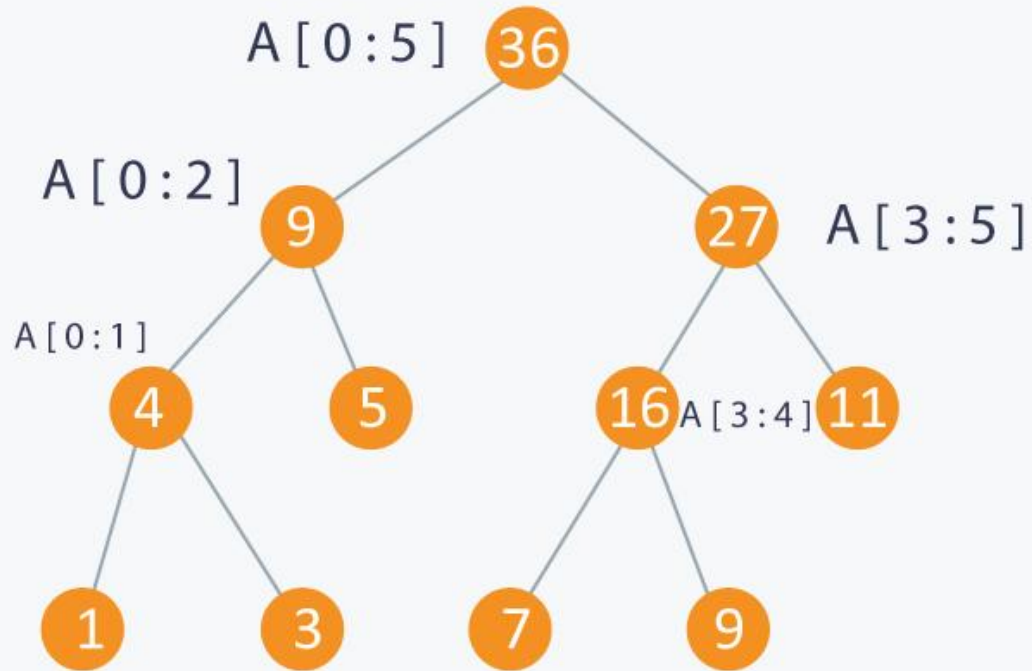
Dado un Arreglo A , Se desea realizar Q queries de la forma “L R “ donde tengo que devolver :

$$\sum A[i] \quad L \leq i \leq R$$

Pero que pasa si agregamos un tipo de query nuevo de la forma “i v“ donde tengo que hacer:

$$A[i] = v$$

- Un segment tree es un árbol binario completo que tiene una operación **asociativa** y dicha operación tiene que tener un neutro.
- Un segment tree puede contener otras cosas que no sean únicamente valores
- Las hojas del árbol representan los elementos del arreglo , por lo tanto hay N hojas
- Los nodos intermedios representan intervalos de la forma $[i,j]$ donde $0 \leq i < j \leq N$



Segment Tree for A = {1,3,5,7,9,11}

Construcción

No se construye como un grafo, pero lo podemos ver como un arreglo donde la raíz es la posición 1 y los hijos del nodo i se ubican en las posiciones $2*i$ y $2*i + 1$.

El tamaño de este arreglo va a ser lo suficientemente grande como para contener todos los nodos que representan intervalos necesarios ,es decir, $4*N$

```

#define oper min
#define NEUT INF
struct STree { // segment tree for min over
integers
    vector<int> st;int n;
    STree(int n): st(4*n+5,NEUT), n(n) {}
    void init(int k, int s, int e, int *a){
        if(s+1==e){st[k]=a[s];return;}
        int m=(s+e)/2;
        init(2*k,s,m,a);init(2*k+1,m,e,a);
        st[k]=oper(st[2*k],st[2*k+1]);
    }
    void upd(int k, int s, int e, int p, int v){
        if(s+1==e){st[k]=v;return;}
        int m=(s+e)/2;
        if(p<m) upd(2*k,s,m,p,v);
        else upd(2*k+1,m,e,p,v);
        st[k]=oper(st[2*k],st[2*k+1]);
    }
}

```

```

int query(int k, int s, int e, int a, int b){
    if(s>=b||e<=a)return NEUT;
    if(s>=a&&e<=b)return st[k];
    int m=(s+e)/2;
    return
oper(query(2*k,s,m,a,b),query(2*k+1,m,e,a,b));
}
void init(int *a){init(1,0,n,a);}
void upd(int p, int v){upd(1,0,n,p,v);}
int query(int a, int b){return query(1,0,n,a,b);}
}; // usage: STree
rmq(n);rmq.init(x);rmq.upd(i,v);rmq.query(s,e);

```

https://github.com/mhunicken/icpc-team-notebook-el-vasito/blob/master/data_structures/segment_tree.cpp

```
void build(int *a, int node = 1, int start = 1, int end = MAXN){  
    if(start == end) tree[node] = a[start-1];  
    else{  
        int mid = (start+end)/2;  
        build(a, 2*node, start, mid);  
        build(a, 2*node+1, mid+1, end);  
        tree[node] = op(tree[2*node], tree[2*node+1]);  
    }  
}
```

Updates

```
void update(int pos, int value, int node = 1, int start = 1, int end = MAXN){
    int mid = (start+end)/2;
    if(start == end){ // estoy en una hoja
        tree[node] = value;
    }else{
        if(start <= pos && pos <= mid){// el indice está del lado izquierdo
            update(pos, value, 2*node, start, mid);
        }else{// el indice está del lado derecho
            update(pos, value, 2*node+1, mid+1, end);
        }
        tree[node] = op(tree[2*node], tree[2*node+1]);
    }
}
```


Queries

```
int query(int l, int r, int node = 1, int start = 1, int end = MAXN){  
    // intervalo del nodo completamente afuera  
  
    if(start > r || end < l) return NEUT;  
  
    // intervalo del nodo completamente adentro  
  
    if(l <= start && end <= r) return tree[node];  
  
    int mid = (start+end)/2;  
  
    // recursión sobre los hijos  
  
    return op(query(l, r, 2*node, start, mid), query(l, r, 2*node+1, mid+1, end));  
}
```

¿Qué complejidad tiene la query sobre segment tree?

- Notemos que en cada nivel del árbol se expande a lo sumo dos nodos. Podemos ver que es absurdo que se expande tres o más veces.
- Entonces se realizan a lo sumo $2 \cdot \log(N)$ operaciones, por lo que la complejidad de la query es $O(\log(N))$

¿Cómo usarlo?

```
int main(){
    cin >> n;
    fore(i,0,n)cin>>a[i];
    STree st(n);
    st.init(a);
    int res = st.query(l,r); // intervalo []
    st.upd(pos,value)
    return 0;
}
```

Problema

<https://codeforces.com/group/KwBzUrrGrr/contest/224389/problem/D>

longest increasing subsequence (LIS)

Recordemos el ejercicio F del contest de DP:

<https://codeforces.com/group/KwBzUrrGrr/contest/244155/problem/F>

Problema de Tarea

<https://www.spoj.com/problems/DQUERY/>