Histogram(Recursion)

Prob:maximum area rectangle:

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long
struct st
{
   ll mx,pos;
} tree[100005 * 4];

ll arr[100005];

ll N;

void build(ll node,ll b, ll e)
{

   if(b == e)
   {
      tree[node].mx = arr[b];
      tree[node].pos = b;
      return;
   }
   ll right = node *2;
   ll left = node* 2 + 1;
   ll mid = (b + e) / 2;

   build(left, b,mid);
   build(right,mid + 1,e);

   if(tree[left].mx<tree[right].mx)
   {
      tree[node].mx = tree[left].mx;
      tree[node].pos = tree[left].pos;
   }
   else
   {
      tree[node].mx = tree[right].mx;
      tree[node].pos = tree[right].pos;
   }
}

st query(ll node, ll b , ll e, ll i , ll j)
{
   //cout<<"haha"<<endl;
   if(b>j || e<i)
   {
      st x;
      x.mx = 99999999999;
      x.pos = 9999999999;
      return x;
   }
   if(b>=i && e<=j)
   {
      return tree[node];
   }
   ll right = node *2;
   ll left = node* 2 + 1;
   ll mid = (b + e) / 2;
   st l = query(left,b,mid,i,j);
   st r = query(right,mid + 1,e,i,j);
   st w;
   if(l.mx<r.mx)
   {
      w.mx = l.mx;
      w.pos = l.pos;
   }
   else
   {
      w.mx = r.mx;
      w.pos = r.pos;
   }
   return w;
}
ll fun(ll beg, ll End)
{
   if(beg>End) return 0;

   if(beg == End)
   {
```

```
        return arr[beg];
    }
    if(beg+1 == End)
    {
        st val = query(1,1,N,beg,End);
        return val.mx * 2;

    }
    st val = query(1,1,N,beg,End);
    ll ans1 = 0,ans2 = 0;
    ans1 = max(val.mx * (End - beg + 1),max(fun(beg,val.pos-
1),fun(val.pos+1,End)));
    return ans1;
}
int main()
{
    ll t;
    scanf("%lld",&t);

    while(t--)
    {
        scanf("%lld",&N);
        for(int i=1; i<=N; i++)
        {
            scanf("%lld",&arr[i]);

        }

        build(1,1,N);
        //     ll Ans = fun(1,N);
        cout<<"query"<<endl;
        ll Q;
        cin>>Q;
        while(Q--)
        {
            ll frm,to;
            cin>>frm>>to;
            st Ans = query(1,1,N,frm,to);
            printf("frm %lld to > %lld  pos = %lld\n",frm,to,Ans.pos);
            printf("rec = %lld\n",Ans);
        }
    }

    return 0;
}
```