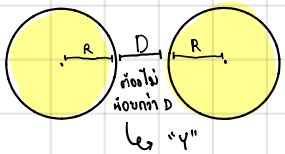
Closert pair segment tree

Closest pair of point $O(n \log n)$

Plantation _ TOI 14



ต้องไม่น้อยกว่า D

↳ "Y"

จุดศูนย์กลางต้องห่างกันไม่น้อยกว่า 2R+D

① Sort จุดตามแกน x

② Divide (แบ่งตามจำนวน)

③ conquer } เหลือจุดไม่เกิน 3 จุด

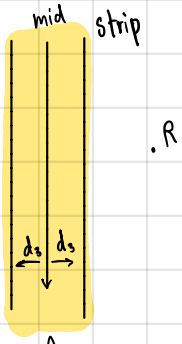
closest (l, r)

mid = (l+r)>>1 ← (l+r)/2 เร็วกว่า

d_1 = closest (l, mid)

d_2 = closest (mid+1, r)

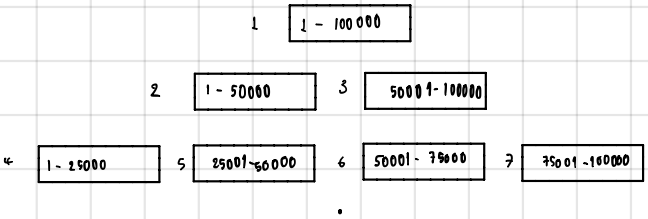
d_3 = min(d_1, d_2)



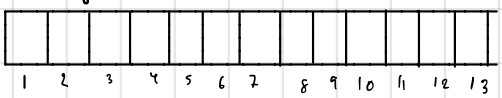
④ sort strip ตาม y

Brute-Force check ระยะห่างไม่เกิน d_3 จุด

Segment tree



int seg[]



process ของ segment tree

① update

② query

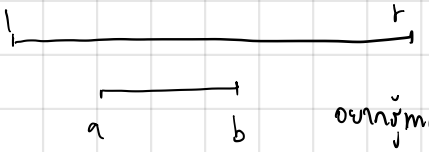
การ build / construct tree

```
void build (int l, int r, int now)
{   if(l==r){
        seg[now] = ค่าของข้อมูลนั้น(a[l])
        return ;
    }
    mid = (l+r)/2 ;
    build (l, mid, now*2);
    build (mid+1, r, now*2+1);
    seg[now] = max (seg[now*2], seg[now*2+1]);
}
```

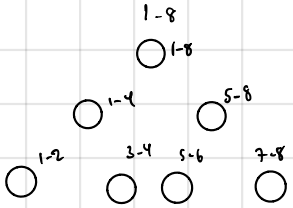
ใน main

```
build(1, n, 1)
```

```
void update (int l, int r, int a, int b, int now)
{   if(l>a || r<a) return ;   เช็คขอบ
    if(l==r){
        seg[now] += b; return ;
    }
    mid = (l+r)/2;
    update (l, mid, a, b, now*2)
    update (mid+1, r, a, b, now*2+1)
    seg[now] = max (seg[now*2], seg[now*2+1])
}
```



ตอบคำถาม max จาก a ถึง b (query)




```
if(l>a || r<b)
    return -1e9;
if(l==r)
    return seg[now];
else
    printf("%d\n",
query(1, n, a, b, 1);
```