

The K-th Rectangle

출제자: 노세윤
정해: 우선순위 큐

풀이

x_1, x_2, \dots, x_N 과 y_1, y_2, \dots, y_M 을 각각 오름차순으로 정렬합시다.

$a_i = x_{i+1} - x_i (1 \leq i < N)$, $b_j = y_{j+1} - y_j (1 \leq j < M)$ 으로 정의합니다.

모든 $a_i \cdot b_j$ 중에서 K 번째로 작은 수가 정답입니다.

이제 구체적인 알고리즘을 설명하겠습니다.

우선 b_1, b_2, \dots, b_{M-1} 을 오름차순으로 정렬한 뒤 우선순위 큐(min-heap)에 $a_1 \cdot b_1, a_2 \cdot b_2, \dots, a_{N-1} \cdot b_1$ 을 넣습니다.

그 뒤 다음 시행을 $K - 1$ 번 반복합니다.

- 우선순위 큐에서 top 원소를 제거합니다.
- 제거한 원소가 $a_i \cdot b_j (j < M - 1)$ 라면 $a_i \cdot b_{j+1}$ 을 우선순위 큐에 넣습니다.

시행이 모두 끝난 뒤 우선순위 큐의 top 원소가 정답이 됩니다.

시간복잡도는 $O((N + K) \cdot \log N + M \cdot \log M)$ 입니다.

K 의 값에 관계없이 파라메트릭 서치를 이용해 $O(N \cdot (\log N + \log M \cdot \log 10^{18}) + M \cdot \log M)$ 으로 푸는 방법도 있습니다.

```
#include <bits/stdc++.h>
using namespace std;
int n, m, k;
long long a[100005];
long long b[100005];
int x[100005];
int y[100005];

struct A{
    int q, w;
    bool operator <(const A&r) const{
        return a[q]*b[w] > a[r.q]*b[r.w];
    }
};
int main(){
    cin.tie(0);
    ios_base::sync_with_stdio(0);
```

```

cin>>n>>m;
int i;
for(i=0 ; i<n ; i++)
    cin>>x[i];
for(i=0 ; i<m ; i++)
    cin>>y[i];
sort(x,x+n);
sort(y,y+m);
for(i=1 ; i<n ; i++)
    a[i-1]=(x[i]-x[i-1]);

for(i=1 ; i<m ; i++)
    b[i-1]=(y[i]-y[i-1]);

sort(a,a+n-1);
sort(b,b+m-1);
priority_queue<A> Q;
for(i=0 ; i<n-1 ; i++)
    Q.push({i,0});
cin>>k;
for(i=1 ; i<k ; i++){
    A t=Q.top();
    Q.pop();
    if(t.w+1<m-1)
        Q.push({t.q,t.w+1});
}
cout<<a[Q.top().q]*b[Q.top().w];
return 0;
}

```