## Problema Proiectoare

**Autor** *Adrian Budău*
Universitatea București,
Facultatea de matematică și informatică

## Descrierea soluției

În primul și în primul rând, trebuie observat că niciodată nu merită folosit un proiector **[u, v]** inclus intr-un alt proiector [a, b], adica **`a <= u <= v <= b`** astfel, folosind o stivă, se pot obține toate intervalele neincluse în alte intervale mai lungi.

**Cazul K = 1**
Se poate observa că răspunsul pentru query-ul **`[x, y]`** este fie un prefix de forma **`[x,z]`**, cu **`z <= y`**, fie un sufix **`[z, y]`**, cu **`x <= z`**, fie este complet inclus, adică **`[u, v]`**, cu **`x < u < v < y`**.
Pentru a determina răspunsul prefix (și analog sufix) se poate cauta binar în intervalele rămase după eliminarea de mai sus intervalul care are capătul din stânga cel mai mare, dar încă mai mic sau egal cu x, acest interval având indicele i1 după ordonare. Analog, pentru suffix, se obtine indicele i2, evident cu **`i1 <= i2`**.
Rămâne astfel de tratat cazul în care răspunsul este un interval complet inclus în **`[x, y]`** și se poate observa foarte ușor că toate intervalele cu indici de la **`i1 + 1`** la **`i2 - 1`** respectă această proprietate și dintre acestea il vom alege pe cel mai lung. Problema se reduce la intrebari de forma: "*care este valoarea maxima a unei valori din intervalul (***`i1 + 1`***,* **`i2 - 1`***), unde valorile sunt lungimile intervalelor* ?"
Aceste intrebari se pot rezolva "offline" folosind divide et impera. La un pas (begin, end) cu **`mid = (begin + end) / 2`** se rezolvă recursiv toate întrebările complet incluse în **`(begin, mid)`** și **`(mid + 1, end)`**, iar apoi toate întrebările care conțin elemente din ambele jumătăți, folosind o preprocesare de maxim partial.
Complexitatea acestei solutii este **`O(N log N)`** pentru sortarea intervalelor, plus **`O(Q log N)`** pentru determinarea întrebărilor ce trebuiesc puse, plus **`O((Q + N) log N)`** pentru determinarea răspunsurilor folosind *divide et impera,* deci complexitatea finală este **`O((Q + N) log N).`**

**Cazul K = 2**
Se poate observa că, odata stabilit că se folosește un proiector **`[u1, v1]`** ca solutie într-un query cu K **`= 2`**, iar acest proiector este cel mai din stânga ales, atunci clar cel mai din dreapta ales este acel proiector **`[u2, v2]`** cu **`u2`** maxim și **`u2 <= v1`**, indiferent cum arată query-ul. Astfel, se poate transforma orice proiector **`[u1, v1]`** în **`[u1, v2]`** cu **`v2`** din definiția de mai sus. Pentru asta se folosesc indecși care se incrementează în parallel, iar după această transformare rezolvarea este identică cu K = 1. Complexitatea este tot **`O((Q + N) log N)`**.

**Cazul `K <= 30`**
Se poate aplica rationamentul de la **`K = 2`** de **`K`** ori si astfel complexitatea este **`O(N * K + (Q + N) log N)`**

**Cazul `K <=` N**
Se poate iarăși aplica raționamentul de la **`K <= 30`**, dar cu ideea de la exponențierea rapidă. Astfel, dacă avem **`K`** impar, se extinde un interval obținut din **`K - 1`** reuniuni, cu alt interval, altfel se extinde un interval obținut din **`K / 2`** reuniuni tot cu unul obtinut din **`K / 2`** reuniuni. Complexitatea finală este **`O(N log K + (Q + N) log N)`**.