

# Problema retenție

Autor Petru Trîmbițaș - WellCode

## Soluție 1 - programare dinamică - $N * K * \log K + N * K * P + N * P * P$ - 100 puncte

Pentru fiecare utilizator vom reuni intervalele care se intersectează obținând un șir de intervale disjuncte și vom sorta aceste intervale.

Pentru fiecare dintre cei  $N$  utilizatori vom calcula  $\text{maxUR}[i][op]$  - retenția maximă care poate fi obținută din primele  $i$  intervale folosind  $op$  operații.

Pentru a calcula  $\text{maxUR}[i][op]$  ne interesează numărul de operații necesare pentru a lega intervalul  $i$  de intervalul  $i - 1$ , fie el  $cop$ .

$$cop = a[i] - b[i - 1] - 1.$$

$$\text{Deci } \text{maxUR}[i][op] = \text{maxUR}[i-1][op-cop] + b[i] - b[i - 1]$$

Acum vom calcula  $\text{userCost}[i][op]$  - retenția maximă care poate fi obținută pentru utilizatorul  $i$  folosind  $op$  operații. Acest lucru îl vom calcula parcurgând toate intervalele  $i$  pentru un anumit utilizator.

Având calculat  $\text{userCost}$  ultima parte a problemei devine foarte asemănătoare cu problema rucsacului.

Pentru obținerea punctajului maxim este necesară tratarea cazurilor particulare precum cazul în care avem un singur utilizator care se loghează într-un singur interval.