**Problema 2 – Ksecv – Descriere soluție**

**Autori Eugenie Daniel Posdarascu, Mihai Ciucu, Radu Voroneanu**

O(**N** \* **K** \* **K**) – 20 de puncte

Programare dinamică

Best[i][j] = Maximul dacă se împart primele i numere în j grupe.

O(**N** log **ValMax**) –40 de puncte

Se cauta binar rezultatul, facand o functie de verificare greedy in felul urmator:

Pentru fiecare secventa noua, incercam sa ne intindem cat mai mult la drepta, cat timp conditia (MaxCurent – MinCurent + 1) / 2 <= valoare. Dacă putem împărți astfel șirul în K sau mai puține subsecvențe, updatăm soluția curentă și scădem valoarea căutată. Alftel, creștem valoarea căutată.

O(**N** + log **ValMax** \***K** \* log2**N**) – 40 de puncte

Rafinăm soluția de mai sus:

Construim în O(**N**) doi arbori de intervale, unul ținând minimul valorilor pe interval, altul ținând maximul valorilor. Cu ajutorul acestora putem să facem fiecare din cele **K** testări de extindere a intervalului curent la dreapta căutând binar capătul subsecvenței cu care extinedm, și testând maximul și minimul pe subsecvența cu care am dori să extindem.

O(**N** + log **ValMax** \* **K** \* log**N**) – 100 de puncte

Îmbunătățim și mai mult soluția de mai sus dacă în loc de căutare binară, căutăm direct pe arborii de intervale secvența cu care extindem.

O(**N** + log **ValMax** \* (**N**/**B** + min(**K**\***B**, **N**)), cu **B** ales optim (ciucuială) – 100 de puncte

Consideram vectorul împărțit în blocuri de câte **B** elemente. Reținem minimul și maximul pentru fiecare bloc, și facem o îmbunătățire a soluției în O(**N** log **ValMax**)

Dacă ne aflăm cu elementul curent la începutul unui bloc, încercăm să adăugăm câte un bloc întreg la secvența curentă. Deoarece împărțim în maxim K subsecvențe, va trebui să parcurgem element cu element maxim K blocuri, deci per total **K\*B** elemente, iar pe restul le vom sări. **B** trebuie ales încât **N/B = K\*B**, deci **B =sqrt(N/K)**.