

Domaća zadaća 5

Strukture podataka i algoritmi I - I053

11. studenoga 2022.

Upute za rješavanje domaće zadaće

Domaća zadaća se predaje preko Teams platforme. Rok predaje je 16. studenog 2022. u 11:59. Za predaju teorijskih zadatak možete skenirati rješenja ili ih natipkati u \LaTeX -u i sve ih zajedno predati u jednoj pdf datoteci. Programerski zadaci se predaju kao .cpp datoteke.

Zadatak 1 (10 bodova). Dano je polje A od n brojeva i dana je veličina prozora k . Za svaku poziciju $i = 1 \dots n - k$ izračunajte $\sum_{j=i}^{i+k} A[j]$. Svoje rješenje implementirajte kao funkciju

```
vector<int> ksusm(const vector<int> &A, int k)
```

koja mora imati vrijeme izvršavanja $\Theta(n)$. Implementirajte i funkciju

```
vector<int> kxors(const vector<int> &A, int k)
```

koja umjesto sume u funkciji iznad računa xor (exclusive or ili \wedge u C++) elemenata prozora. Ova funkcija se isto treba izvršavati u vremenu $\Theta(n)$.

Zadatak 2 (20 bodova). Kada Matko ima slobodnog vremena ide u knjižnicu čitati knjige. Danas ima t minuta slobodnog vremena. Zato je Matko uzeo n knjiga iz knjižnice i za svaku procijenio koliko mu treba vremena da ju pročita. Za knjigu i mu treba $A[i]$ minuta da ju pročita.

Matko je odlučio izabrati jednu knjigu i od nje će početi čitati. Ako je izabrao knjigu i , onda će pročitati knjige i , $i + 1$, $i + 2$ itd. sve dok mu ne istekne vrijeme ili dok ne pročita zadnju knjigu. Ukoliko mu istekne slobodno vrijeme u pola čitanja knjige, onda tu knjigu neće brojati kao pročitano.

Pomozite Matku da izračuna koliko najviše knjiga može pročitati ako izabere optimalnu početnu knjigu.

Dan je vektor A veličine $1 \leq n \leq 2 \cdot 10^5$ i slobodno vrijeme $1 \leq t \leq 10^9$. Vrijeme čitanja knjige će biti prirodan broj $1 \leq A[i] \leq 10^4$. Svoje rješenje implementirajte kao funkciju

```
int books(const vector<int> &A, int t)
```

koja mora imati vrijeme izvršavanja $O(n \log n)$. (Hint: a može se riješiti i u $\Theta(n)$).

Zadatak 3 (20 bodova). U hotel se u isto vrijeme prijavilo n turista a samo je m slobodnih apartmana. Vaš zadatak je rasporediti apartmane tako da što više turista dobije apartman. Svaki turist ima željenu veličinu apartmana i prihvatit će apartman samo ako se njegova veličina ne razlikuje za više od k od željene. Odnosno ako turist želi apartman veličine x , onda će on prihvatiti apartman ne manji od $x - k$ i ne veći od $x + k$.

Dano je polje za turiste A veličine $1 \leq n \leq 10^5$, polje za apartmane B veličine $1 \leq m \leq 10^5$ i prirodan broj $1 \leq k \leq 10^9$ iz zadatka. Želje turista će biti prirodni brojevi $1 \leq A[i] \leq 10^9$ i veličine apartmana će biti brojevi $1 \leq B[i] \leq 10^9$. Implementirajte funkciju koja

```
int apartments(const vector<int> &A, const vector<int> &B, int k)
```

koja vraća koliko je turista dobilo apartman u optimalnoj raspodijeli apartmana, u vremenu $O(n \log n)$.

Zadatak 4 (20 bodova). Profesor Karnak tvrdi da kako je FORESEE optimalan algoritam koji predviđa budućnost, onda on na svakom koraku ne smije imati veću cijenu nego MOVE-TO-FRONT algoritam. Formalno dokažite ili opovrgnite protuprimjerom profesorovu tvrdnju.

Zadatak 5 (10 bodova). Originalna pretpostavka za *swap* u MOVE-TO-FRONT algoritmu je da *swap* operacija košta 1 i mijenja mjesta susjednim elementima liste. Relaksirajmo tu operaciju na način da nakon što pristupimo elementu x , mi smijemo po trošku 1 zamijeniti element x sa bilo kojim elementom prije njega. Pokažite da sa tako relaksiranom operacijom *swap*, algoritam MOVE-TO-FRONT je 2-kompetitivan.