Priprema za kolokvijum: Strukture podataka

Pravila:

- Rešenja se pišu u Python-u ili u pseudokodu.
- Nije dozvoljeno korišćenje ugrađenih funkcija koje obavljaju posao iz zadatka (`sort`, `sorted`, `min`, `max`, `index`, `reverse` i slične). Dozvoljeni su `len`, `range` i `print`.
- Kod lančanih listi menjaju se pokazivači, a ne podaci u čvorovima, osim ako zadatak kaže drugačije.
- Klase čvora, liste, steka i reda nije potrebno pisati, osim metoda koje zadatak traži.

Zadaci

Zadatak 1: Binarno traženje sa ponavljanjima (20 poena)

a) Dato je rastuće sortirano polje `a` u kome se vrednosti mogu ponavljati. Napisati funkciju `count_occurrences(a, x)` koja vraća koliko puta se `x` pojavljuje u polju, sa složenošću O(log n): pomoću dva binarna traženja, jednog za prvo i jednog za poslednje pojavljivanje. Linearni prolaz kroz polje se ne priznaje. (15 poena)

b) Za polje `[1, 2, 2, 2, 3, 5, 8]` i `x = 2` prikazati vrednosti `low`, `high` i `mid` u svakom koraku traženja prvog pojavljivanja. (5 poena)

Zadatak 2: Gornje trougaona matrica (15 poena)

Gornje trougaona matrica reda n ima nule ispod glavne dijagonale. Čuvaju se samo elementi na dijagonali i iznad nje, red po red, u jednodimenzionalnom polju.

a) Izvesti formulu za poziciju elementa (i, j), j ≥ i, u tom polju. Za n = 5 izračunati koliko se elemenata čuva i na kojoj poziciji je element (2, 4). Prikazati postupak. (6 poena)

b) Napisati klasu `UpperTriangularMatrix` sa metodama `get(i, j)` i `put(i, j, value)`. Čitanje ispod dijagonale vraća 0, a upis vrednosti različite od nule ispod dijagonale podiže izuzetak. Indeksi van matrice takođe podižu izuzetak. (9 poena)

Zadatak 3: Brisanje svih pojavljivanja (25 poena)

Data je jednostruko ulančana lista sa pokazivačima `head` i `tail`. Napisati metodu `delete_all(el)` klase `SLList` koja iz liste briše **sva** pojavljivanja vrednosti `el` i vraća broj obrisanih čvorova. Posle brisanja `head` i `tail` moraju biti ispravni, i u slučaju kada lista postane prazna.

Primer: lista `3 → 7 → 3 → 3 → 5 → 3`, poziv `delete_all(3)` vraća 4, a lista postaje `7 → 5`.

Zadatak 4: Najveći element na kraj (20 poena)

Napisati metodu `move_max_to_end()` klase `SLList` koja pronalazi čvor sa najvećom vrednošću i premešta ga na kraj liste, **prevezivanjem pokazivača** (ne zamenom vrednosti `info`). Ako ima više najvećih, premešta se prvi. Lista se ne obilazi više od jednom radi traženja.

Primer: `9 → 4 → 1 → 6` postaje `4 → 1 → 6 → 9`.

Zadatak 5: Obrtanje prvih k elemenata reda (20 poena)

Napisati funkciju `reverse_first_k(q, k)` koja obrće redosled prvih `k` elemenata reda `q`, dok ostali elementi zadržavaju svoj redosled. Dozvoljeno je koristiti samo operacije reda (`enqueue`, `dequeue`, `is_empty` i atribut `size`, broj elemenata) i **jedan pomoćni stek** (`push`, `pop`, `is_empty`). Ako `k` nije između 0 i broja elemenata reda, funkcija podiže izuzetak.

Primer: red `1 2 3 4 5 6` (početak levo) i `k = 3` daju red `3 2 1 4 5 6`.