# Strukture podataka i algoritmi

## Projektni zadatak 009\_FLIGHTS

Vaš zadatak je napisati program (konzolnu aplikaciju u C++) koji će korisniku omogućiti rad s podacima o aerodromima. Program mora biti smisleno strukturiran i intuitivan. U nastavku su opisane funkcionalnosti koje treba program imati. Za izradu projekta se koristi datoteka naziva **SPA\_PROJ\_009\_FLIGHTS\_data\_2.csv**:

* + IATA\_CODE: Location Identifier
  + AIRPORT: Airport's Name
  + CITY: City Name of the Airport
  + STATE: State Name of the Airport
  + COUNTRY: Country Name of the Airport
  + LATITUDE: zemljopisna širina
  + LONGITUDE: zemljopisna dužina

1. (Bodova: **1**) Pokretanjem aplikacije prikažite korisniku sljedeći izbornik:

Pretraga po STATE

Prikaz po zemljopisnoj širini

Kopiranje u novi vektor

Prikaz po zemljopisnoj širini i dužini

1. (Bodova: **2**) Dizajnirajte i implementirajte tip podataka Airport koji može čuvati sve podatke o aerodromima iz datoteke **SPA\_PROJ\_009\_FLIGHTS\_data\_2.csv**. Pripremite tip podataka tako da se može konstruirati na smislene načine te definirajte gettere, settere i potrebne metode za svakog člana. Omogućite korisniku da pozivanjem metode na objektu može ispisati osnovne podatke o aerodromu na ekran.
2. (Bodova: **2**) Kad korisnik odabere „Pretraga po STATE“, prvo prekopirajte sve aerodrome iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše STATE pa mu nakon toga ispišite podatke o svim aerodromima iz te države. Osigurajte da se pretraživanje odvija u logaritamskoj složenosti koristeći najbolji od sljedećih kontejnera: map, multimap, set, multiset.
3. (Bodova: **2**) Promijenite prethodni zadatak tako da pretraživanje radite prvo po odabranom kontejneru, a zatim i po vektoru i po listi (kopirajte podatke i u listu). Ispišite top ljestvicu brzine kontejnera: prvo prikažite najbrži, onda srednji pa najsporiji.
4. (Bodova: **2**). Kad korisnik odabere „Prikaz po zemljopisnoj širini“, koristeći prioritetni red ispišite aerodrome padajuće prema zemljopisnoj širini.
5. (Bodova: **3**) Kad korisnik odabere „Kopiranje u novi vektor“, kreirajte novi vektor i napravite sljedeće:

* Prekopirajte sve aerodrome u novi kontejner koji omogućuje da se pretraživanje po gradu odvija u logaritamskoj složenosti (birajte map, multimap, set ili multiset).
* Sve dok to korisnik želi, omogućite mu unos grada.
* Kad korisnik unese neki grad, pronađite sve aerodrome iz tog grada u vašem odabranom kontejneru i kopirajte ih u novi vektor.
* Ponavljajte sve dok to korisnik želi.
* Na kraju ispišite sve aerodrome iz novog vektora.

1. (Bodova: **3**) Kad korisnik odabere „Prikaz po zemljopisnoj širini i dužini“, koristeći prioritetni red ispišite sve aerodrome rastuće prema zemljopisnoj širini. Sve aerodrome na istoj zemljopisnoj širini obavezno prikažite sortirane rastuće prema zemljopisnoj dužini.
2. (Bodova: **1**) Dodajte u izbornik sljedeće opcije:

Bubble vs Shell

Pogodi trajanje

Lets Bogo!

The Merge Dance

1. (Bodova: **3**) Kad korisnik odabere „Bubble vs Shell“, napravite sljedeće:

* Kreirajte dva pomoćna vektora *v1* i *v2* i iskopirajte u svako od njih sve aerodrome.
* Uzmite s predavanja Bubble i Shell sortiranja te ih promijenite tako da sortiraju aerodrome, padajuće po IATA šifri.
* Sortirajte vektor *v1* koristeći Bubble sort.
* Sortirajte vektor *v2* koristeći Shell sort.
* Ispišite koliko je trajalo svako sortiranje.

1. (Bodova: **2**) Promijenite prethodni zadatak tako da svaki algoritam pri sortiranju zbraja broj zamjena mjesta (*swap*) koje napravi i vraća to kao izlaznu vrijednost iz funkcije. Sortirajte pa ispišite za oba sortiranja broj zamjena mjesta.
2. (Bodova: **3**) Kad korisnik odabere opciju „Pogodi trajanje“, generirajte jedan slučajni broj *n* između 1 i 100.000.000 te pitajte korisnika da upiše koliko milisekundi misli da će trajati razbacivanje, sortiranje i binarno pretraživanje tog polja u potrazi za brojem 7. Kad korisnik upiše vrijednost, generirajte to polje s vrijednostima od 1 do *n*, razbacajte ga, sortirajte ga te binarnim pretraživanjem i pronađite broj 7. Ispišite korisniku kolika je razlika između njegove pogođene vrijednosti trajanja i konkretnog trajanja.
3. (Bodova: **4**) Kad korisnik odabere „Lets Bogo!“, slučajnim odabirom prekopirajte pet aerodroma u novi vektor. Isprogramirajte svoju vlastitu determinističku verziju bogo sorta te je iskoristite za slaganje aerodroma po IATA šifri u rastućem redoslijedu. Ispišite u tekstualnu datoteku svaku isprobanu permutaciju.
4. (Bodova: **4**) Kad korisnik odabere opciju „The Merge Dance“, pitajte ga da upiše broj *n* koji predstavlja broj elemenata. Nakon što upiše broj, u vektor ubacite slučajnim redoslijedom *n* pokemona. Zatim vektor sortirajte Merge sortom (koristite i promijenite prema potrebi implementaciju s predavanja) te za vrijeme sortiranja radite sljedeće:

* Svaki put kad se napravi poziv rekurzivne funkcije, brojač povećajte za 1
* Svaki put kad se završi poziv rekurzivne funkcije, brojač smanjite za 1.
* Za svaku promjenu brojača iscrtajte u tekstualnu datoteku jedan redak koji sadrži onoliko znakova # kolika je trenutna vrijednost brojača.

1. (Bodova: **1**) Dodajte u izbornik sljedeće opcije:

Pretraga po STATE

Vizualiziraj

Izrada stringa iz stringa

1. (Bodova: **2**) Kad korisnik odabere „Pretraga po STATE“, prvo prekopirajte aerodrome iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše vrijednost koju želi tražiti pa mu nakon toga ispišite podatke o svim aerodromima iz te savezne države. Osigurajte da se pretraživanje odvija u najboljoj mogućnoj složenosti koristeći neki od sljedećih kontejnera: unordered\_map, unordered\_multimap, unordered\_set, unordered\_multiset.
2. (Bodova: **2**) Promijenite prethodni zadatak tako da pretraživanje radite prvo po odabranom kontejneru, a zatim i po vektoru i po listi (kopirajte podatke i u listu). Ispišite top ljestvicu brzine kontejnera: prvo prikažite najbrži, onda srednji pa najsporiji.
3. (Bodova: **5**) Kad korisnik odabere „Vizualiziraj“, natrpajte sve aerodrome u unordered\_map tako da ključ bude njihova IATA šifra. Nakon toga, vizualizirajte *hash* tablicu iz unordered\_multimap tako da iscrtate sve njene *buckete* te uz svaki *bucket* prikažete i aerodrome u njemu. Primjer početka ispisa (vrijednosti su izmišljene):

Bucket 0: ACT -- COS -- GRB

Bucket 1: HDN -- LAS -- MMH -- SMF

Bucket 2: [EMPTY]

1. (Bodova: **6**) Kad korisnik odabere „Izrada imena iz imena“, omogućite mu da odabere jedan aerodrom. Koristeći unordered\_map, unordered\_multimap, unordered\_set ili unordered\_multiset, ispišite može li se IATA šifra odabranog aerodroma kreirati koristeći raspoloživa slova iz punog naziva aerodroma (pri tome mala i velika slova smatrajte jednakima). Primjerice:

* Ako je puno ime aerodroma „Wiley Post-Will Rogers Memorial Airport“, a IATA šifra „BRW“, onda se IATA šifra ne može konstruirati iz imena aerodroma jer nedostaje slovo b.
* Ako je puno ime aerodroma „Fort Wayne International Airport“, a IATA šifra „FWA“, onda se IATA šifra može konstruirati iz imena aerodroma.