# Strukture podataka i algoritmi

## Projektni zadatak 011\_IMDB

Vaš zadatak je napisati program (konzolnu aplikaciju u C++) koji će korisniku omogućiti rad s podacima o filmovima na IMDB-u. Program mora biti smisleno strukturiran i intuitivan. U nastavku su opisane funkcionalnosti koje treba program imati. Za izradu projekta se koristi datoteka **SPA\_PROJ\_011\_IMDB\_data.csv**:

* Rank: Movie rank order
* Title: The title of the film
* Genre: A comma-separated list of genres used to classify the film
* Description: Brief one-sentence movie summary
* Director: The name of the film's director
* Actors: A comma-separated list of the main stars of the film
* Year: The year that the film released as an integer.
* Runtime (Minutes) : The duration of the film in minutes.
* Rating: User rating for the movie 0-10
* Votes: Number of votes
* Revenue (Millions) : Movie revenue in millions
* Metascore: An aggregated average of critic scores. Values are between 0 and 100. Higher scores represent positive reviews.

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

Pretraga po Metascoreu

Prikaz po godini

Kopiranje u novi vektor

Prikaz po generacijama

1. (Bodova: **2**) Dizajnirajte i implementirajte tip podataka Movie koji može čuvati sve podatke o filmu iz datoteke. 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 filmu na ekran (naziv, opis, godina izdanja, zarada u hrvatskom formatu, primjerice „29.002.000,00 USD“).
2. (Bodova: **3**) Kad korisnik odabere „Pretraga po Metascoreu“, prvo prekopirajte sve filmove iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše min i max Metascore ocjene pa mu nakon toga ispišite podatke o svim filmovima u tom rasponu. 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 godini“, koristeći prioritetni red ispišite filmove padajuće prema godini.
5. (Bodova: **5**) Kad korisnik odabere „Kopiranje u novi vektor“, kreirajte novi vektor i napravite sljedeće:

* Prekopirajte sve filmove u novi kontejner koji omogućuje da se pretraživanje po prvom slovu naslova odvija u logaritamskoj složenosti (birajte map, multimap, set ili multiset).
* Sve dok to korisnik želi, omogućite mu unos prvog slova naslova.
* Kad korisnik unese neko slovo, pronađite sve filmove koji počinju tim slovom u vašem odabranom kontejneru i premjestite ih u novi vektor.
* Ponavljajte sve dok to korisnik želi.
* Na kraju ispišite sve filmove iz novog vektora.

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

Selection vs Insertion

Pogodi trajanje

Lets Bogo!

The Merge Dance

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

* Kreirajte dva pomoćna vektora *v1* i *v2* i iskopirajte u svakog od njih sve filmove.
* Uzmite s predavanja Selection i Insertion sortiranja te ih promijenite tako da sortiraju filmove, rastuće po nazivu.
* Sortirajte vektor *v1* koristeći Selection sort.
* Sortirajte vektor *v2* koristeći Insertion 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 filmova u novi vektor. Isprogramirajte svoju vlastitu determinističku verziju bogo sorta te je iskoristite za slaganje filmova po nazivu u rastućem redoslijedu. Ispišite na ekranu 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* filmova. Zatim vektor sortirajte Merge sortom po nazivu (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 žanru

Vizualiziraj

Izrada stringa iz stringa

1. (Bodova: **2**) Kad korisnik odabere „Pretraga po žanru“, prvo prekopirajte filmove iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše žanr koji želi tražiti pa mu nakon toga ispišite podatke o svim filmovima tog žanra. 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 filmove u unordered\_multimap tako da ključ bude godina. Nakon toga, vizualizirajte *hash* tablicu iz unordered\_multimap tako da iscrtate sve njene *buckete* te uz svaki *bucket* prikažete i filmove u njemu. Primjer početka ispisa (vrijednosti su izmišljene):

Bucket 0: Prometheus -- Their Finest -- Ex Machina

Bucket 1: Twilight -- Iron Man -- Children of Men -- Snowpiercer

Bucket 2: [EMPTY]

1. (Bodova: **6**) Kad korisnik odabere „Izrada imena iz imena“, omogućite mu da odabere jedan film i upiše neki string. Koristeći unordered\_map, unordered\_multimap, unordered\_set ili unordered\_multiset, ispišite može li se uneseni string kreirati koristeći raspoloživa slova iz opisa filma (pri tome mala i velika slova smatrajte jednakima). Primjerice:

* Ako je opis filma „Following clues to the origin of mankind, a team finds a structure on a distant moon, but they soon realize they are not alone“, a uneseni string „fog“, onda se uneseni string može konstruirati iz opisa filma.
* Ako je opis filma „Following clues to the origin of mankind, a team finds a structure on a distant moon, but they soon realize they are not alone.“, a uneseni string „xog“, onda se uneseni string može konstruirati iz opisa filma jer u opisu nedostaje slovo x.