# Strukture podataka i algoritmi

## Projektni zadatak 004\_NETFLIX

Vaš zadatak je napisati program (konzolnu aplikaciju u C++) koji će korisniku omogućiti rad s filmovima i serijama u ponudi Netflixa. Program mora biti smisleno strukturiran i intuitivan. U nastavku su opisane funkcionalnosti koje treba program imati. Za izradu projekta se koristi datoteka **SPA\_PROJ\_004\_NETFLIX\_data.csv**. Opis stupaca:

* show\_id: Unique ID for every Movie / TV Show
* type: Identifier - A Movie or TV Show
* title: Title of the Movie / TV Show
* director: Director of the Movie
* cast: Actors involved in the movie / show
* country: Country where the movie / show was produced
* date\_added: Date it was added on Netflix
* release\_year: Actual Release year of the move / show
* rating: TV Rating of the movie / show
* duration: Total Duration - in minutes or number of seasons
* listed\_in: Genre
* description: The summary description

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

Pretraga po godini izdanja

Prikaz po godini izdanja

Kopiranje u novu listu

Prikaz po vrsti

1. (Bodova: **2**) Dizajnirajte i implementirajte tip podataka MovieOrShow koji može čuvati sve podatke o filmu/seriji iz datoteke **SPA\_PROJ\_004\_NETFLIX\_data.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 filmu/seriji na ekran (ime, godina produkcije, žanr, opis).
2. (Bodova: **2**) Kad korisnik odabere opciju „Pretraga po godini izdanja“, prvo prekopirajte sve filmove/serije iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše godinu izdanja koju želi tražiti pa mu nakon toga ispišite podatke o svim filmovima/serijama izdanim te godine (svojstvo „release\_year“) ili napišite da nema niti jednog. 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 izdanja“, pitajte korisnika da odabere prikaz od novijih prema starijim ili obratno (svojstvo „release\_year“) te želi li filmove ili serije. Koristeći prioritetni red, ispišite korisniku sve tražene filmove ili serije, poštujući odabrani redoslijed.
5. (Bodova: **3**) Kad korisnik odabere „Kopiranje u novu listu“, kreirajte novu listu *l* i napravite sljedeće:

* Prekopirajte sve filmove/serije u novi kontejner koji omogućuje da se pretraživanje po svojstvu „show\_id“ odvija u logaritamskoj složenosti (optimalno birajte map, multimap, set ili multiset).
* Sve dok to korisnik želi, omogućite mu unos vrijednosti za „show\_id“.
* Kad korisnik unese vrijednost, pronađite odgovarajući film/seriju u vašem odabranom kontejneru pa ih prekopirajte ih u listu *l*.
* Na kraju ispišite sadržaj liste *l*.

1. (Bodova: **3**) Kad korisnik odabere „Prikaz po vrsti“, koristeći prioritetni red ispišite sve filmove/serije prema vrsti, prvo filmove, a zatim serije. Sve filmove obavezno prikažite padajuće prema broju država iz svojstva „country“, zatim isto ponovite i za serije.
2. (Bodova: **1**) Dodajte u izbornik sljedeće opcije:

Selection vs Shell

Lets play guess

Monkey Play

Counting

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

* Kreirajte dva pomoćna vektora *v1* i *v2* i iskopirajte u svako od njih sve filmove/serije.
* Uzmite s predavanja Selection i Shell sortiranja i promijenite ih tako da sortiraju filmove/serije, padajuće po naslovu.
* Sortirajte vektor *v1* koristeći Selection 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 „Lets play guess“, generirajte jedan slučajni broj *n* između 1 i 100.000.000 te pitajte korisnika da upiše koliko mikrosekundi misli da će trajati razbacivanje, sortiranje i binarno pretraživanje tog vektora u potrazi za brojem 13. Kad korisnik upiše vrijednost, generirajte taj vektor s vrijednostima od 1 do *n*, razbacajte ga, sortirajte ga te binarnim pretraživanjem i pronađite broj 13. Ispišite korisniku kolika je razlika između njegove pogođene vrijednosti trajanja i konkretnog trajanja.
3. (Bodova: **4**) Kad korisnik odabere „Monkey Play“, slučajnim odabirom prekopirajte tri filma/serije u novi vektor. Isprogramirajte svoju vlastitu slučajnu verziju bogo sorta te je iskoristite za slaganje filmova/serija po nazivu u rastućem redoslijedu. Ispišite koliko je pokušaja bilo potrebno da se elementi sortiraju.
4. (Bodova: **4**) Kad korisnik odabere opciju „Counting“, pitajte ga da upiše broj *n* koji predstavlja maksimalni element. Nakon što upiše broj, u vektor ubacite 100.000 slučajnih brojeva između 1 i *n*. Zatim vektor sortirajte Counting sortom (koristite i promijenite prema potrebi implementaciju s predavanja) te za vrijeme sortiranja radite sljedeće:

* Prilikom prebrajanja koliko ima kojih elemenata, za svaki obrađeni element ispišite u tekstualnu datoteku sadržaj pomoćnog polja koje sadržava broj pronađenih elemenata. Primjerice, ako se radi o polju [ 2, 2, 1, 3, 2 ], u datoteci treba pisati:

0 0 0 0

0 0 1 0

0 0 2 0

0 1 2 0

0 1 2 1

0 1 3 1

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

Pretraga po imenu

Vizualiziraj

Izrada imena iz imena

1. (Bodova: **2**) Kad korisnik odabere „Pretraga po imenu“, prvo prekopirajte filmove/serije iz vektora u odgovarajući kontejner. Zatim pitajte korisnika da upiše ime koje želi tražiti pa mu nakon toga ispišite podatke o filmu/seriji s tim imenom ili ispišite da ne postoji. 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/serije u unordered\_map tako da ključ bude njihov show\_id. Nakon toga, vizualizirajte *hash* tablicu iz unordered\_map tako da iscrtate sve njene *buckete* te uz svaki *bucket* prikažete i filmove/serije u njemu. Primjer početka ispisa (vrijednosti su izmišljene):

Bucket 0: Good People -- Lovesick -- Catwoman

Bucket 1: Leap! -- Pawn Stars -- The Forgotten -- Resurface

Bucket 2: [EMPTY]

1. (Bodova: **6**) Kad korisnik odabere „Izrada imena iz imena“, omogućite mu da odabere dva filma/serije te mu zatim, koristeći unordered\_map, unordered\_multimap, unordered\_set ili unordered\_multiset, ispišite može li se ime drugog filma/serije kreirati koristeći raspoloživa slova iz imena prvog filma/serije (pri tome mala i velika slova smatrajte jednakima). Primjerice:

* Ako je ime prvog filma/serije „Killer Owl“, a drugog „Loki“, onda se ime drugog može konstruirati iz prvog.
* Ako je ime prvog filma/serije „Lady Bullseye“, a drugog „Lady Bulls Eye“, onda se ime drugog ne može konstruirati iz prvog jer nedostaje jedan razmak.