# Sveučilište u Zagrebu

# Prirodoslovno-matematički fakultet

# Matematički odsjek

# Završni projekt– Multimedijski sustavi

# Aplikacija za vizualizaciju sortiranja

Iris Trgovec

Zagreb, veljača 2025.

#### ****Opis projekta****

Projekt predstavlja interaktivnu aplikaciju za vizualizaciju popularnih algoritama sortiranja, kao što su **Bubble Sort**, **Selection Sort**, **Insertion Sort** i **Quick Sort**. Aplikacija omogućava korisnicima unos vlastitog niza brojeva, odabir algoritma za sortiranje te praćenje rada algoritama u stvarnom vremenu.

Koristeći grafičke prikaze, aplikacija vizualizira usporedbe i zamjene između brojeva tijekom sortiranja, čime korisnici mogu pratiti rad algoritma korak po korak. Također, aplikacija nudi mogućnosti pauziranja, ubrzavanja i usporavanja sortiranja, kao i ponovnog pokretanja aplikacije. Aplikacija omogućava dinamičan i edukativan način razumijevanja osnovnih algoritama za sortiranje.

#### ****Tehnologije korištene u projektu****

**Razvojno okruženje**: Processing IDE

**Biblioteke**: Processing (vizualizacija)

**Literatura**

<https://web.math.pmf.unizg.hr/nastava/prog1/materijali/predavanja-skracenoMM/Predavanje13.pdf> [zadnje pristupljeno 6.02.2025.]

<https://www.geeksforgeeks.org/insertion-sort-algorithm/> [zadnje pristupljeno 6.02.2025.]

<https://www.geeksforgeeks.org/selection-sort-algorithm-2/> [zadnje pristupljeno 4.02.2025]

#### ****Dnevnik rada na projektu****

| Opis zadatka | Broj utrošenih sati | Napomena |
| --- | --- | --- |
| 1. Postavljanje osnovne strukture aplikacije i sučelja | 6 | Postavljanje temeljnog okvira aplikacije, inicijalizacija osnovnih varijabli, osnovno grafičko sučelje. |
| 2. Implementacija funkcionalnosti za unos brojeva i prikaz niza | 6 | Omogućavanje korisnicima unos brojeva, prikaz unesenih brojeva na ekranu. |
| 3. Implementacija algoritma Bubble Sort | 6 | Programiranje i testiranje Bubble Sort algoritma. |
| 4. Izrada vizualizacije usporedbi i zamjena | 5 | Razrada grafičkog prikaza usporedbe brojeva, vizualizacija zamjena tijekom sortiranja. |
| 5. Implementacija Selection Sorta | 6 | Programiranje i testiranje Selection Sort algoritma. |
| 6. Implementacija Insertion Sorta | 6 | Programiranje i testiranje Insertion Sort algoritma. |
| 7. Dodavanje gumba za ubrzavanje, usporavanje i pauzu | 5 | Implementacija kontrole brzine sortiranja, opcija pauze i nastavka sortiranja. |
| 8. Dodavanje Quick Sorta i funkcionalnosti za ponovno pokretanje aplikacije | 7 | Programiranje Quick Sort algoritma, dodavanje gumba za resetiranje i ponovni unos niza. |
| 9. Testiranje aplikacije | 7 | Testiranje svih algoritama, pronalaženje i ispravljanje bugova. |
| 10. Poboljšanje korisničkog sučelja i optimizacija prikaza podataka | 3 | Povećanje vizualne učinkovitosti aplikacije, optimizacija prikaza za veće nizove brojeva kod sortiranja. |
| 11. Pisanje dokumentacije i dnevnik rada | 3 | Pisanje dokumentacije o projektu, dnevnik rada. |

Ukupno utrošeno vremena na projekt: 60 sati

#### ****Funkcionalnosti aplikacije****

* **Unos niza brojeva**: Korisnik može unijeti niz brojeva razdvojenih zarezima, koji se zatim prikazuje u aplikaciji.
* **Odabir algoritma sortiranja**: Korisnik može odabrati jedan od četiri algoritma sortiranja: Bubble Sort, Selection Sort, Insertion Sort i Quick Sort.
* **Vizualizacija algoritama**: Za svaki odabrani algoritam, aplikacija prikazuje korak po korak usporedbe i zamjene brojeva.
* **Kontrola brzine**: Korisnik može ubrzati ili usporiti proces sortiranja (od 0.5x do 5x).
* **Pauza i nastavak**: Sortiranje može biti pauzirano i nastavljeno kad god korisnik poželi.
* **Ponovno pokretanje**: Aplikacija omogućuje resetiranje aplikacije i početak s novim unosom brojeva.

#### ****Implementacija****

Aplikacija koristi **Processing IDE** za razvoj grafičkog sučelja i animacija. Glavna funkcionalnost aplikacije temelji se na vizualizaciji algoritama sortiranja kroz usporedbe i zamjene. Za implementaciju algoritama koristile su se standardne inačice svakog algoritma:

* **Bubble Sort**: Jednostavan algoritam koji uspoređuje susjedne elemente i zamjenjuje ih ako nisu u pravom redoslijedu.
* **Selection Sort**: Algoritam koji pronalazi najmanji element i zamjenjuje ga s trenutnim elementom na početku niza.
* **Insertion Sort**: Algoritam koji "ubacuje" elemente na odgovarajuće pozicije u već sortiranom dijelu niza.
* **Quick Sort**: Podjela niza na manje dijelove oko odabranog pivota i rekurzivno sortiranje tih dijelova.