

# **STRUKTURE PODATAKA I ALGORITMI**

Vježbe 11

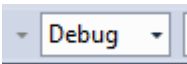
# Zadaci

1. Implementirajte varijantu bogo sorta kojom rastuće sortiranje radite tako da prolazite sve permutacije.
2. Učitajte prvih 20.000 slučajnih brojeva iz datoteke `puno_brojeva1.txt` u program. Primijenite bubble, insertion, selection, shell i merge iz priloga na brojeve iz datoteke i ispišite za svaki koliko je trajao.
3. Promijenite prethodni zadatak tako da koristite prvih 20.000 brojeva iz datoteke `puno_brojeva2.txt` (brojevi su već sortirani kako treba – najbolji mogući slučaj).
4. Promijenite prethodni zadatak tako da koristite prvih 20.000 brojeva iz datoteke `puno_brojeva3.txt` (brojevi su sad sortirani padajuće – najgori mogući slučaj).

# Zadaci

5. Promijenite selection sort tako da sortira pravokutnike (širina, visina) prema površini. Učitajte svih 1000 pravokutnika iz pravokutnici.txt (u svakom retku su širina i visina jednog pravokutnika odvojeni razmakom) i prepisite ih u drugu datoteku, ali sortirano.
6. Promijenite insertion sort tako da sortira 2D točke prema njihovoj udaljenosti od ishodišta. Zadaajte nekoliko točaka u programu, sortirajte pa ispišite točke i udaljenosti.

# Zadaci

7. Iz datoteka puno\_brojeva1.txt i puno\_brojeva2.txt učitajte brojeve u dva polja (u svakoj datoteci ima točno milijun brojeva). Primijenite quick sort na svako polje i ispišite koliko je sortiranje trajalo. Koju datoteku sortira dulje i zašto?
  - Promijenite zadatak tako da umjesto Debug konfiguracije koristite Release konfiguraciju (  ). Kakva su sada vremena sortiranja i kako to tumačite? Vratite natrag na Debug konfiguraciju.

8. Učitajte 1.000.000 slučajnih brojeva iz datoteke `puno_brojeva1.txt` u program. Ispišite koliko traje njihovo sortiranje shell sortom. Nakon toga, promijenite shell sort algoritam tako da umjesto originalne sekvence razmaka koristi Tokudinu sekvencu iz 1992. Kako se promijenilo vrijeme sortiranja?
- Uputa za rješavanje: sve sekvence koje počinju od 1 se koriste na sljedeći način: pomoću formule treba unaprijed izračunati sve elemente sekvence manje od veličine polja. Nakon toga, te elemente sekvence treba primijeniti na polje, ali obrnutim redoslijedom. Primjerice, ako imamo polje od 100 elemenata, prvo ćemo izračunati Tokudinu sekvencu 1, 4, 9, 20, 46, a zatim ćemo je primijeniti na polje, ali počevši od 46 pa sve do 1.