

APS 2 — prva domača naloga

Rok za oddajo: nedelja, 8. marca 2026

Časovna analiza algoritmov urejanja

Naloga

V skladu s sledečimi navodili napišite program in narišite diagrame. Oboje oddajte.

Program

V programskem jeziku C++ napišite program za merjenje časa izbranih algoritmov urejanja tabel, ki vsebujejo določeno permutacijo števil od 1 do N . Program napišite tako, da ga bo mogoče pognati na sledeči način:

```
./uredi A N U I
```

Parametri imajo sledeče vrednosti:

- A (algoritem) je lahko 1 (navadno vstavljanje), 2 (urejanje z zlivanjem) ali 3 (bogosort);
- N je dolžina tabele;
- U (začetna urejenost tabele) je lahko 1 (tabela vsebuje zaporedje $1, 2, \dots, N$), 2 (tabela vsebuje zaporedje $N, N-1, \dots, 2, 1$) ali 3 (tabela vsebuje zaporedje $1, N, 2, N-1, 3, N-2, \dots$);
- I je število izvedb postopka.

Program naj I -krat izvede algoritem A na tabeli, kot jo določata parametra N in U , in izpiše povprečno porabo časa (v mikrosekundah) za eno izvedbo urejanja. Povprečno porabo časa seveda izračunate tako, da skupno porabo časa delite z I .

Pazite, da ne boste urejali tabele, ki jo je uredila prejšnja izvedba urejanja! Tabela (ceplusplusovski `vector`, seveda) si pripravite vnaprej, na začetku vsake izvedbe urejanja pa izdelajte njeno kopijo ...

```
vector<int> kopija = tabela;
```

... in jo uredite. V merjenje časa boste tako zajeli tudi čas, potreben za vsakokratno izdelavo kopije tabele, a to je zgolj $O(N)$ z majhno konstanto.

Aha, najbrž vas zanima, kako merimo čas? Preprosto! Na začetek programa dodajte

```
#include <chrono>
```

in

```
using namespace std::chrono;
```

meritev pa izvedete takole:

```

auto zacetek = high_resolution_clock::now();
// ...
// izvedemo tisto, kar merimo
// ...
auto konec = high_resolution_clock::now();
auto trajanje = duration_cast<microseconds>(konec - zacetek);

```

Če boste uporabljali prevajalnik GCC, potem program prevedite s stikalom `-O2` (npr. `g++ -O2 uredi.cpp`).

Diagrami

Izdelajte datoteko `diagrami.pdf`, ki vsebuje devet linijskih diagramov (na vsaki strani po enega). Diagrami naj prikazujejo odvisnost porabe časa (v mikrosekundah) od velikosti tabele (N) pri algoritmu A , začetni urejenosti U in I izvedbah. Parametre določite takole:

- I naj bo pri vseh diagramih enak 100.
- A naj bo pri i -tem diagramu (za $i \in \{1, \dots, 9\}$) enak $\lceil i / 3 \rceil$.
- U naj bo pri i -tem diagramu (za $i \in \{1, \dots, 9\}$) enak $(i - 1) \bmod 3 + 1$.
- Pri $A = 1$ naj N preteče zaporedje $N_0, 2N_0, 3N_0, \dots, 10N_0$, pri čemer N_0 izberite tako, da bo celoten postopek (za vseh 100 ponovitev skupaj) na vašem računalniku pri $U = 2$ in $N = 10N_0$ trajal najmanj 1 minuto in največ 10 minut. Vrednost N_0 tudi izrecno zapišite.
- Pri $A = 2$ ravnajte enako kot pri $A = 1$, verjetno pa bo N_0 tokrat lahko precej večji.
- Pri $A = 3$ in naj N preteče zaporedje $N_0, N_0 + 1, N_0 + 2, \dots, N_0 + 9$, pri čemer N_0 izberite tako, da bo celoten postopek (za vseh 100 ponovitev skupaj) na vašem računalniku pri $U = 2$ in $N = N_0 + 9$ trajal najmanj 1 minuto in največ 10 minut.

Pri $A \leq 2$ naj bo merilo na osi y linearno, pri $A = 3$ pa logaritmčno (1, 10, 100, ...).

Oddaja naloge

Oddajte datoteki `uredi.cpp` in `diagrami.pdf`.