



# ALGORITMI IN PODATKOVNE STRUKTURE I

## Merjenje časa izvajanja

# MERJENJE ČASA IZVAJANJA

- Izmeriti želimo čas izvajanja nekega programa. Problemi...
  - ločljivost funkcije za merjenje
  - v meritve so vključeni časi tudi drugih vzporednih programov,
  - različni časi izvajanja zaradi arhitekture (predpomnilnik),
  - različni vhodni podatki povzročijo različne čase izvajanja



# MERJENJE ČASA IZVAJANJA

```
long zacetniCas = System.nanoTime();
```

```
for (i = 0; i < stPonovitev; i++) {  
    for (j = 0; j < stArgumentov; j++) {  
        //izberi argumente  
        //klic metode z izbranimi argumenti  
    }  
}
```

```
double casIzvajanja = (double)(System.nanoTime() - zacetniCas);  
casIzvajanja /= (stPonovitev * stArgumentov);
```



# OCENA DEJANSKEGA ČASA IZVAJANJA

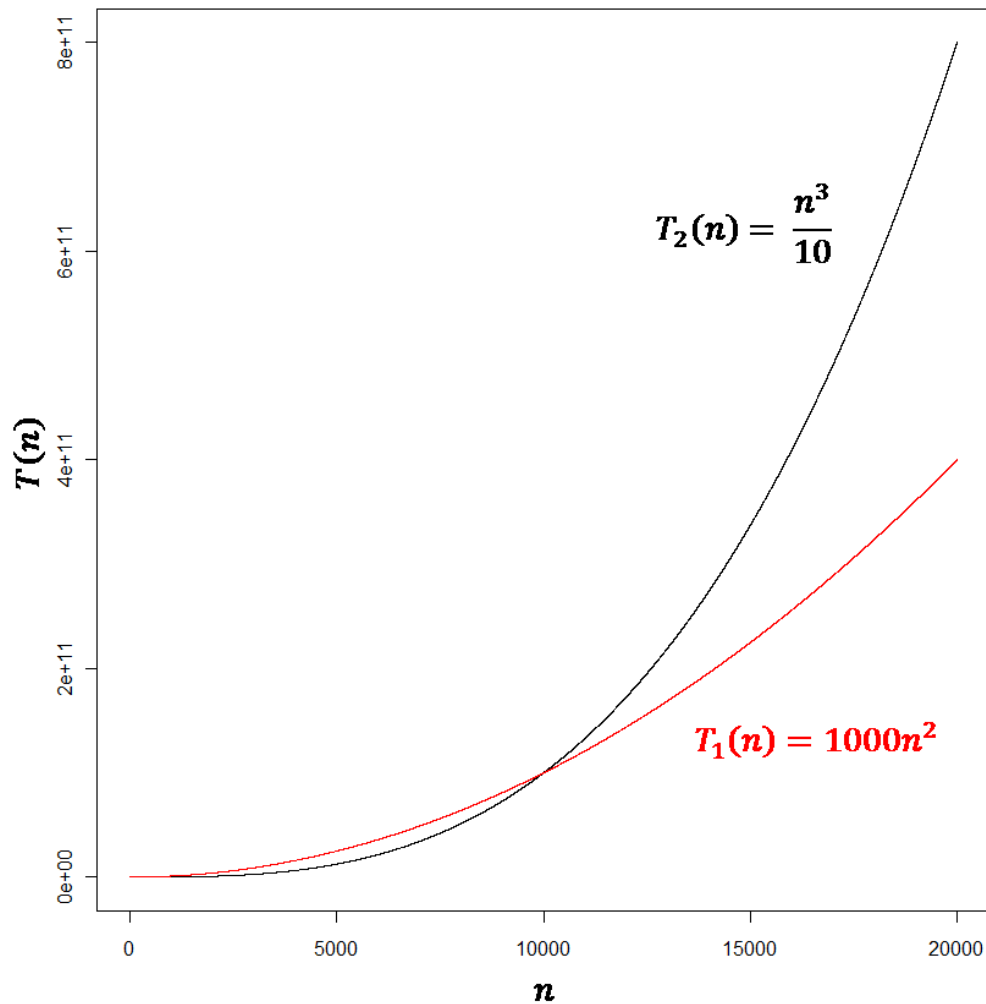
- Pri oceni velikostnega reda časovne zahtevnosti zanemarimo vse člene nižjega reda kot tudi vse konstante
- V realnosti ravno konstante lahko spremenijo sliko uporabnosti algoritma

$$T_1(n) = 1000n^2 \qquad T_2(n) = \frac{n^3}{10}$$

za  $n < 10000$  je drugi algoritem primernejši!



# OCENA DEJANSKEGA ČASA IZVAJANJA



za  $n < 10000$  je drugi algoritem primernejši!



# OCENA DEJANSKEGA ČASA IZVAJANJA

- Predpostavimo neko odvisnost med časovno zahtevnostjo in dejanskim časom izvajanja, npr:

$$T(n) = a * O(g(n)) + c$$

- oceno konstant izvajamo z meritvami
- za določitev konstant v enačbi je potrebno opraviti toliko meritev, kolikor je konstant
- meritve opravljamo pri velikih vrednostih vhodnih parametrov
- rešimo dobljeni sistem enačb
- dobljeno funkcijo lahko uporabimo za napovedovanje časa izvajanja programa