

# Algoritamske strategije

# Predispitne obaveze

- Test – Kompleksnost algoritama (6 poena)
- I kolokvijum – Dinamičko programiranje (20 poena)
- II kolokvijum – Grafovi (20 poena)
- III kolokvijum – Linearno programiranje (20 poena)
- Prisustvo (4 poena)
  
- Predispitne obaveze (70 poena)
- Uslov – najmanje **36 poena**

# Ispit

| predispitne | skalirano | ocena | ispit |
|-------------|-----------|-------|-------|
| 70          | 80.00     | 8     | 9, 10 |
| 69          | 80.00     | 8     | 9, 10 |
| 63          | 80.00     | 8     | 9, 10 |
| 61          | 80.00     | 8     | 9, 10 |
| 60          | 80.00     | 8     | 9     |
| 57          | 80.00     | 8     | 9     |
| 50          | 71.43     | 8     | 8     |
| 49          | 70.00     | 7     | 8     |
| 46          | 65.71     | 7     | 8     |
| 45          | 64.29     | 7     | 8     |
| 43          | 61.43     | 7     | 8     |
| 42          | 60.00     | 6     | 7, 8  |
| 36          | 51.43     | 6     | 7     |

- Zadaci sa takmičenja
  - 10
  - 9
  - do 8
- Dodatna nastava
- Bubble Cup

Kompleksnost algoritama

# Kompleksnost algoritama

- Ograničavajući faktori
  - Brzina izvršavanja
  - Količina memorije
- Veličina ulaza -  $n$
- Vreme izvršavanja – funkcija veličine ulaza,  $f(n)$ , broj koraka u algoritmu za određeni ulaz
- Najgori slučaj izvršavanja algoritma

# Primer 1 - suma elemenata niza

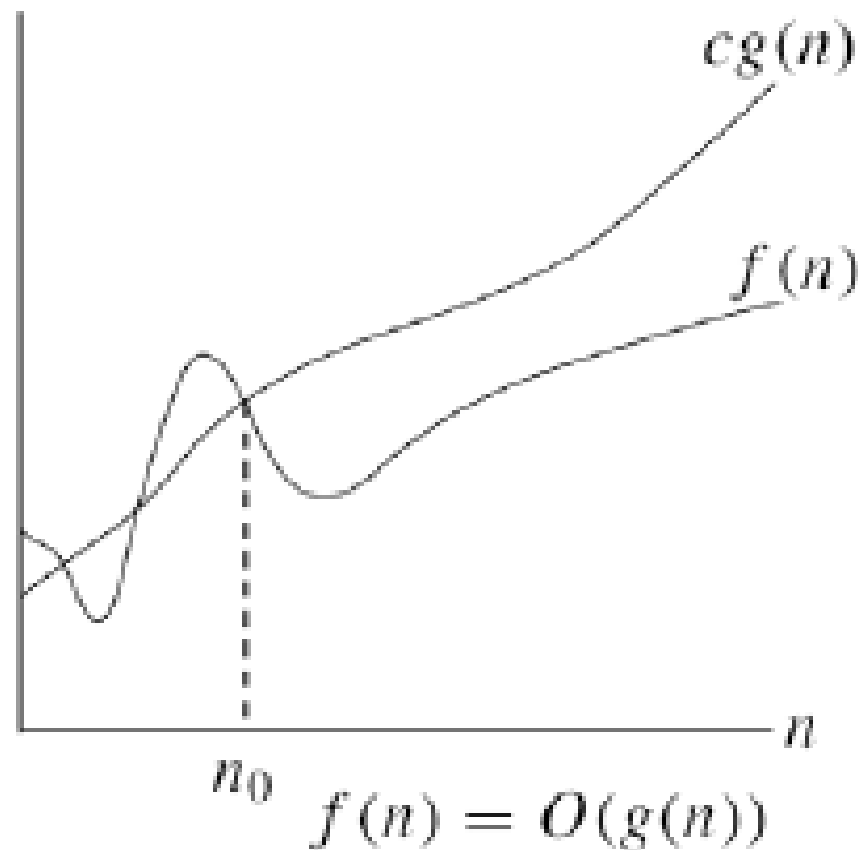
```
Suma(A, n)
  suma = 0
  for i = 1, n
    suma = suma + A(i)
```

- Jedinica mere – operacije dodele vrednosti
- Inicijalizacija – **suma**, **i** (2 instrukcije)
- **n** puta se poveća **suma** i brojač **i** (**2n** instrukcija)
- $f(n) = 2 + 2n = O(n)$
- Linearna kompleksnost, jednostruka petlja

# O-notacija

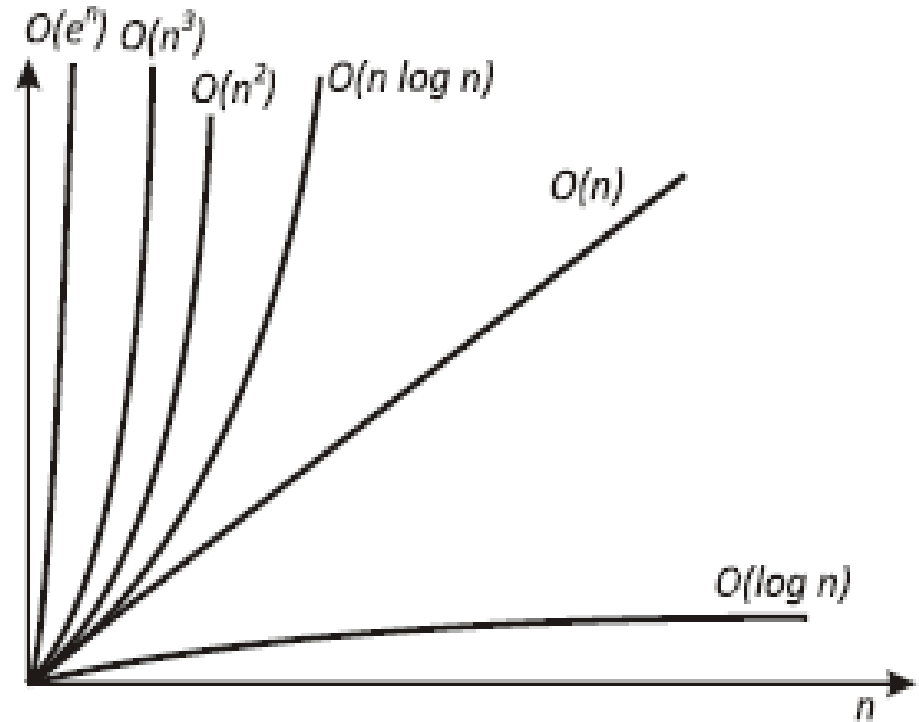
- **Definicija.**

- $f(n) = O(g(n))$
  - kada  $n \rightarrow \infty$ ,
  - ako postoje  $c$  i  $n_0$
  - takvi da je  $f(n) \leq c \cdot g(n)$
  - za svako  $n \geq n_0$ .
- $O(g(n))$  – gornja granica za  $f(n)$
  - Za dovoljno veliko  $c$  i  $n$ ,  $c \cdot g(n)$  uvek postaje veće od  $f(n)$



# O-notacija

- $n^2, 3n^2, n^2 - 5000, 5n^2 + 3n \in O(n^2)$
- $n, 5n, 10n + 6, \log n, n \log n \in O(n^2)$
- $n^3 \notin O(n^2)$





# Primer 2

- Suma svih podnizova koji počinju od prve pozicije
- Jedinica mere – operacije dodele vrednosti i štampanja

```
SumaSub(A, n)
  for i = 1, n
    suma = A(1)
    for j = 2, i
      suma = suma + A(j)
    write(suma)
```

$$\begin{aligned} F(n) &= (n + 1) + n + \sum_{i=1}^n i + \sum_{i=1}^n (i - 1) + n = \\ &= 1 + 3n + \frac{n(n+1)}{2} + \frac{n(n-1)}{2} = 1 + 3n + n^2 = \\ &= O(n^2) \end{aligned}$$

- Dvostruka petlja, kvadratna kompleksnost

# Primer 3

- Suma poslednjih 5 elemenata podnizova koji počinju od prve pozicije
- Jedinica mere – operacije dodele vrednosti (bez štampanja)

```
SumaPos15(A, n)
```

```
  for i = 5, n
```

```
    suma = A(i-4)
```

```
    for j = i-3, i
```

```
      suma = suma + A(j)
```

```
    write(suma)
```

$$\begin{aligned} F(n) &= (n-3) + (n-4) + 5(n-4) + 4(n-4) = \\ &= (n-3) + 10(n-4) = 11n - 43 \\ &= O(n) \end{aligned}$$

- Dvostruka petlja, ali linearna kompleksnost!
- Broj iteracija unutrašnje petlje - konstantan

# Primer 4

- Množenje dve kvadratne matrice  $n \times n$

```
MatrProd(A, B, n)
```

```
  for i = 1, n
```

```
    for j = 1, n
```

```
      C(i,j) = 0
```

```
      for k = 1, n
```

```
        C(i,j) = C(i,j) + A(i,k)*B(k,j)
```

$$F = O(n^3)$$

- Trostruka petlja – kubna kompleksnost

# Primer 5 – Selection sort

- Mera jedinice – broj poređenja u *if*-u

```
Selectionsort(x, n)
  for i=1, n-1
    for j=i+1, n
      if x[j] < x[i] then
        swap(x[j], x[i])
```

$$\begin{aligned} f(n) &= \sum_{i=1}^{n-1} (n-i) = \\ &= (n-1) + (n-2) + \dots + 1 \\ &= \frac{n(n-1)}{2} = \frac{n^2-n}{2} = O(n^2) \end{aligned}$$