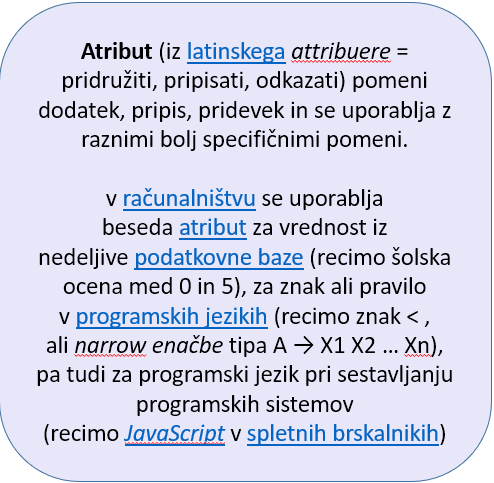
ALGORITMI IN KOMPLEKSNOST

ATRIBTUI ALGORITMOV

Najpomembnejši atributi algoritmov:

* **Pravilnost**(najpomembnejša)
* **Razumljivost**
* **Eleganca**
* **Učinkovitost**

**PRAVILNOST ALGORITMOV**

Vprašati se moramo ali rešujemo pravi problem in ali ga rešujemo pravilno(to se moramo vedno vprašati!)

**RAZUMLJIVOST IN ELEGANCA**

Razumljivost(jasnost algoritmov):

Enostavnost za vzdrževanje in nadgrajevanje. Lažje preverjanje pravilnosti.

Elegenca = lepa rešitev

**Včasih sta si razumljivost in eleganca nasprotujoči, včasih pa ne.**

**UČINKOVITOST ALGORITMOV**

Učinkovitost algoritma = učinkovita izraba virov

* Časovna učinkovitost/potratnost:
  + Merjenje časa izvajanja algoritma
  + Bolj sistematična analiza učinkovitosti algoritmov
* Prostorska učinkovitost/potratnost
  + Količina informacije, ki jo mora algoritem dodatno shraniti

Primerjanje algoritmov (benchmarking):

Merjenje časa izvajanja algoritmov na standardnih množicah podatkov

Primerjanje zmogljivosti računalnikov

Analiza algoritmov: merjenje učinkovitosti algoritmov

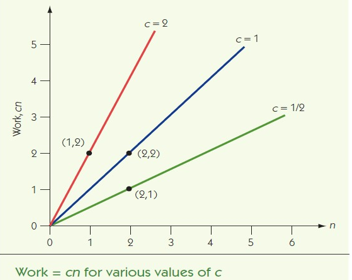
* Primerjamo algoritme, ki opravijo enako nalogo in vrnejo isti rezultat
* Primerjamo časovne kompleksnosti in prostorske kompleksnosti

RED VELIKOSTI – razred n

O(n): množica funkcij, ki rastejo linearno

* Linearna kompleksnost

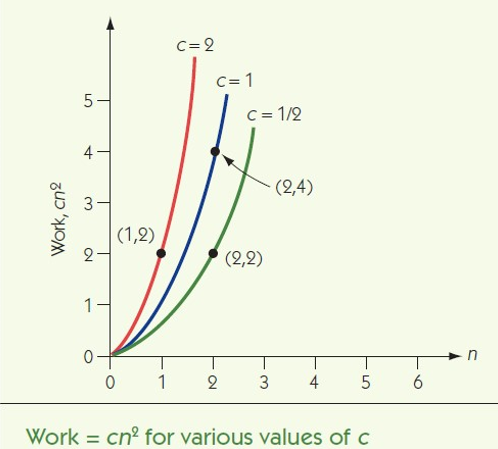
V odvisnosti od velikosti problema (števila vhodnih podatkov)

Sprememba v velikosti je konstanta z večanjem n

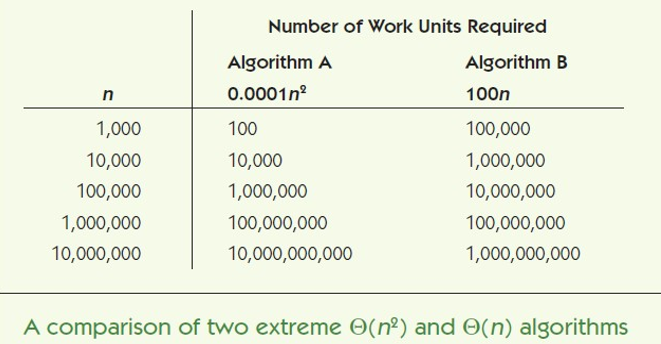
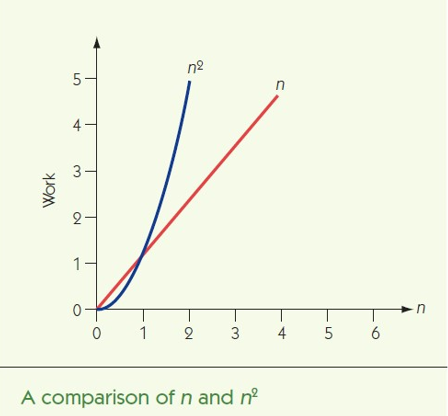
* Red velikosti n

RED VELIKOSTI razred n²

Red velikosti n2: Θ(n2) – algoritem opravi cn2 dela, da sprocesira n elementov



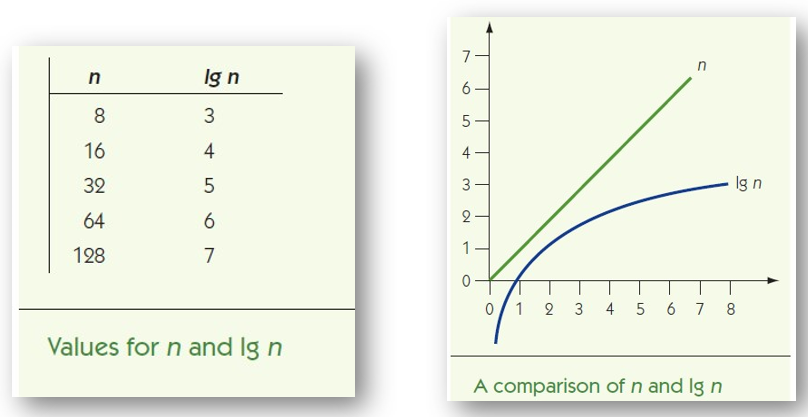
Vsaka funkcija reda velikosti n2 ima od neke točke naprej večje vrednosti od funkcije reda velikosti n:



RED VELIKOSTI – razred log n

Red velikosti log n: Θ(log n) – algoritem opravi le c\*log(n) dela, da sprocesira n elementov

narašča zelo počasi



RED VELIKOSTI – razred kᵑ

Večina algoritmov je polinomsko omejenih (reda nk)

Nekateri niso:

* ne moremo jih rešiti v polinomskem času, niti pri velikih k!

Eksponentna časovna kompleksnost: Θ(kn)

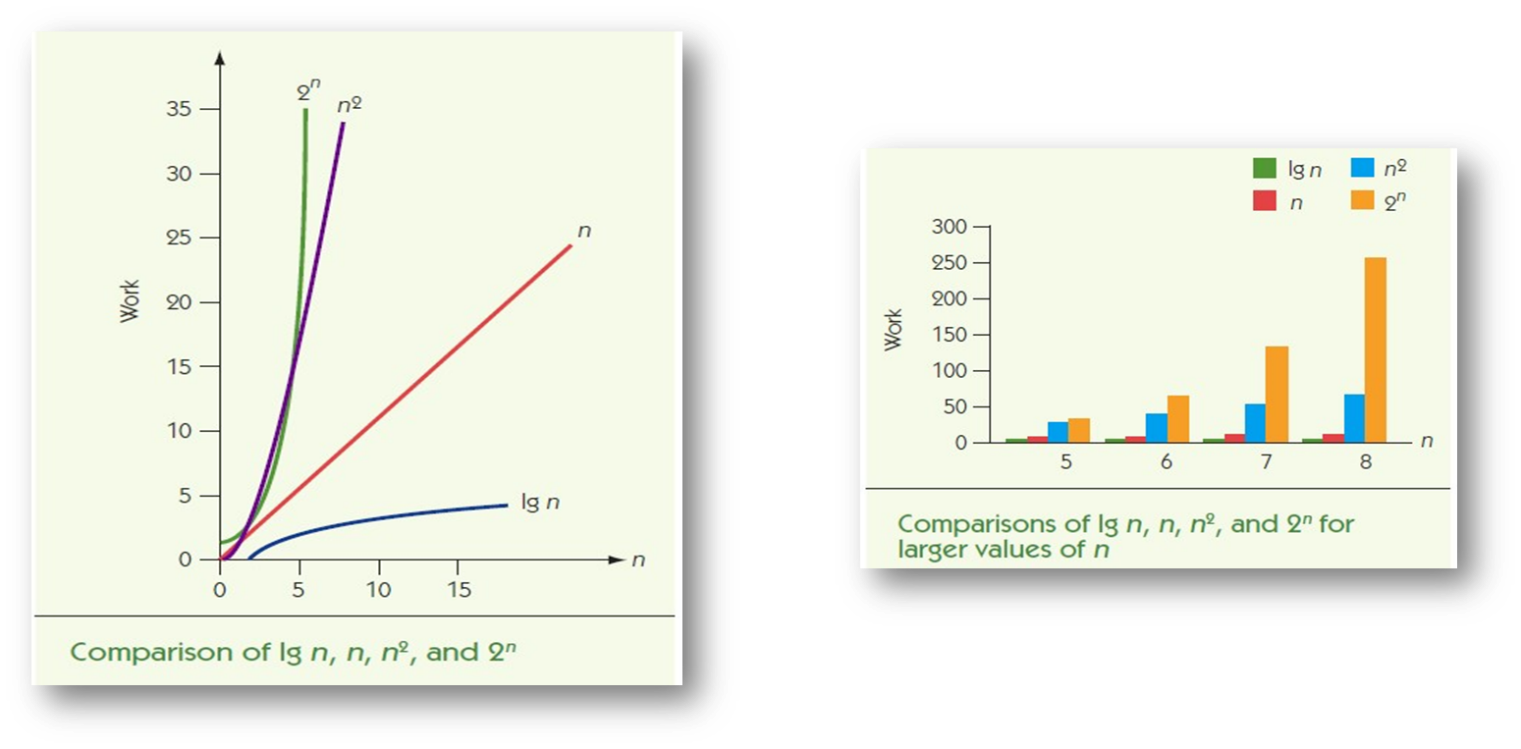
Ogromno število pregledovanj:

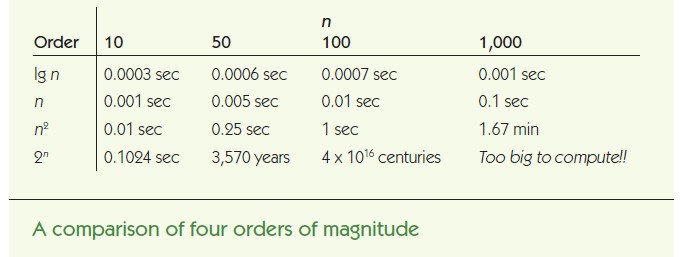
* višina drevesa: n+1
* število listov: 2n= število poti, ki jih moramo pregledati

Problemi ki niso polinomsko omejeni:

* Hamiltonov cikel
* Problem trgovskega potnika
* Polnjenje nahrbtnikov
* Šah

PRIMERJAVE REDOV VELIKOSTI





REDI VELIKOSTI

* V računalništvu poznamo več razredov časovne kompleksnosti:
  + Konstantna O(1)
  + Logaritemska O(log n)
  + „Linearitmična“ O(n \* log n)
  + Linearna O(n)
  + Kvadratična O(n2)
  + Kubična O(n3)
  + Polinomska O(np), p > 0

Eksponentna O(2n)

Pravila pri določanju razredov kompleksnosti

Preštejemo število operacij:

* *Aritmetične operacije*: x+1
* *Primerjave:* x>1
* *Prirejanje*: x=1
* *Klice sistemskih funkcij*: Vnesi x

Produkt: f\*O(g) є O(f\*g)

* + Ponavljanje dela kode
  + Primer: če se n-krat ponovi del kode z O(n), je skupna kompleksnost O(n2)

Vsota: O(f1 + f2) = O(max(f1, f2))

* + Zaporedni segmenti kode
  + Skupna kompleksnost je enaka večji kompleksnosti
  + Pri vejitvah upoštevamo pot z večjo kompleksnostjo

Množenje s konstanto: O(k\*g) = O(g)

* + Množenje s konstanto ne vpliva na razred kompleksnosti

PRIBLIŽNI ALGORITMI

Rešijo problem približno

* Ne zagotavljajo optimalne rešitve
* Običajno pa zagotovijo dovolj dobro rešitev (v spremenljivem času)