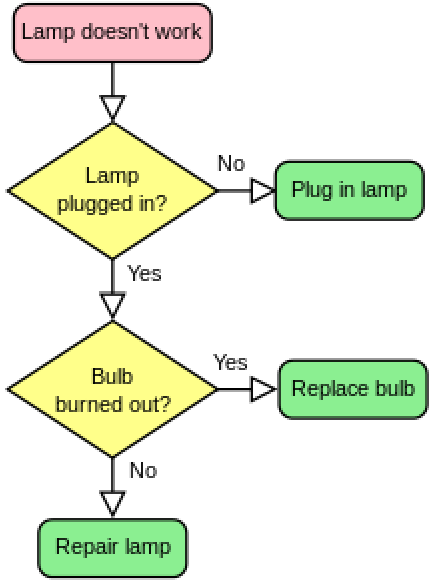
Projektovanje algoritama

L01. Uloga algoritama u računanju

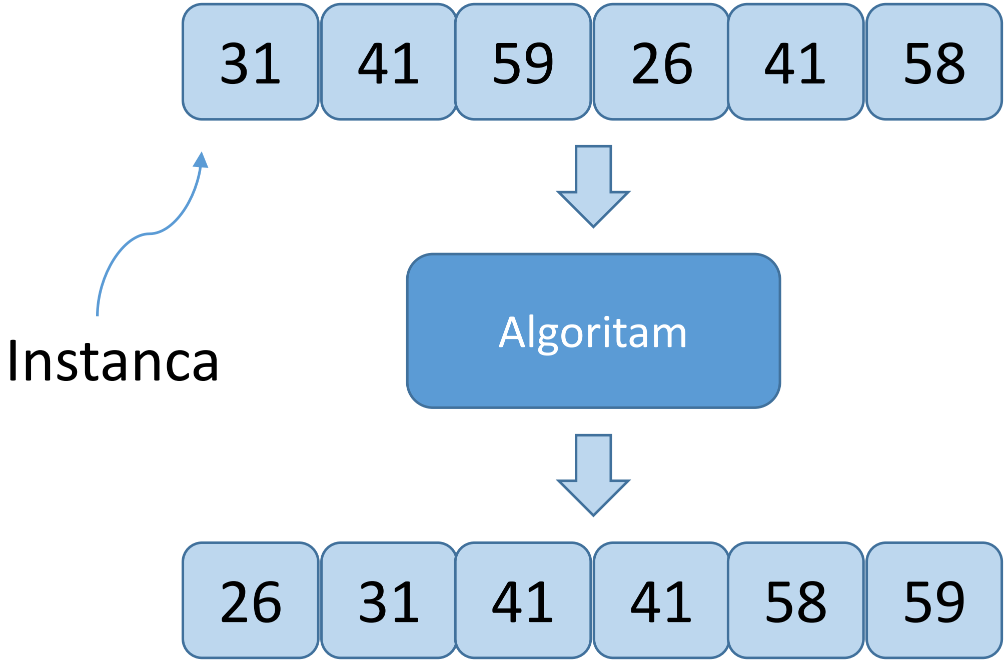
Šta su algoritmi?

Algoritam = bilo koja dobro definisana procedura koja preuzima vrednost ili skup vrednosti kao **ulaz** i formira vrednost ili skup vrednosti kao **izlaz**

Algoritam = alat za rešavanje dobro specificiranog **problema računanja**



Primer algoritma



Problem sortiranja:

Ulaz: niz N brojeva $[a_1, a_2, \ldots, a_N]$

Izlaz: permutacija ulaznog niza $[a'_1, a'_2, \ldots, a'_N]$

takva da je:
$a'_1 <= a'_2 <= \ldots <= a'_N$

Kvalitet algoritma

Algoritam je **tačan** ukoliko se, za **svaku** kombinaciju ulaza, **završava** i formira **tačan izlaz**.

Tačan algoritam **rešava** dati problem računanja.

Od mnogo **tačnih** algoritama cilj je uvek pronaći **najbolji** algoritam za rešenje datog problema računanja, odn. algoritamskog problema.

Nemaju svi algoritamski problemi pronađeno **tačno** rešenje!

Aproksimativan algoritam = dovoljno dobro rešenje datog problema.

Primeri algoritamskih problema

- čuvanje DNK sekvence u bazi podataka
- analiza DNK sekvence i otkrivanje sličnosti

The Human Genome Project

~100,000 gena
~3,000,000,000 baznih parova

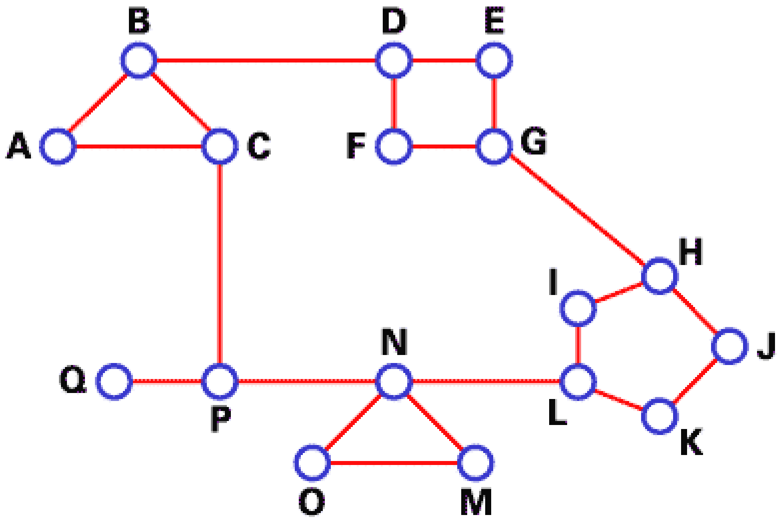
~1,000,000,000,000 (10^{12}) vrsta na Zemlji [1]
(od kojih znamo samo par miliona)

~5.0 x 10^{37} baznih parova (1 par = 2 bita)

→ *Nalaženje najveće zajedničke podsekvence (primer dinamičkog programiranja)*

[1] Staff (2 May 2016). "Researchers find that Earth may be home to 1 trillion species". *National Science Foundation*.

Primeri algoritamskih problema



- komunikacija između dva računara na Internetu

  → *Problem najkraće putanje*

- pretraga Interneta

  → *Brzo pretraživanje pomoću Hash tabela*
  → *Poklapanje stringova*

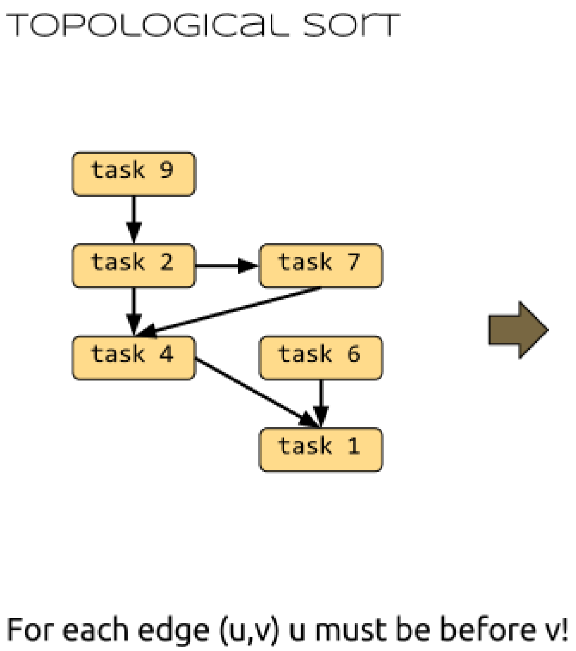
Primeri algoritamskih problema

SHUTTERSTOCK

- elektronsko bankarstvo
- online kupovina
- sigurnost podataka

→ *Algoritmi teorije brojeva*

Primeri algoritamskih problema



Sortiranje, ali <u>ne po aritmetičkom odnosu</u>.

- raspoređivanje zadataka takvo da se svaki zadatak izvrši pre svih zadataka koji koriste rezultat tog zadatka.

→ *Topološko sortiranje*

Strukture podataka

Struktura podataka je način čuvanja i organizacije podataka kako bi se omogućio pristup i modifikacija tih podataka.

→ *Stack, Queue, List*

→ *Hash tabele*

→ *Stabla (Binary Search Tree)*

→ *Grafovi*

„Teški" problemi

Za dati skup lokacija:

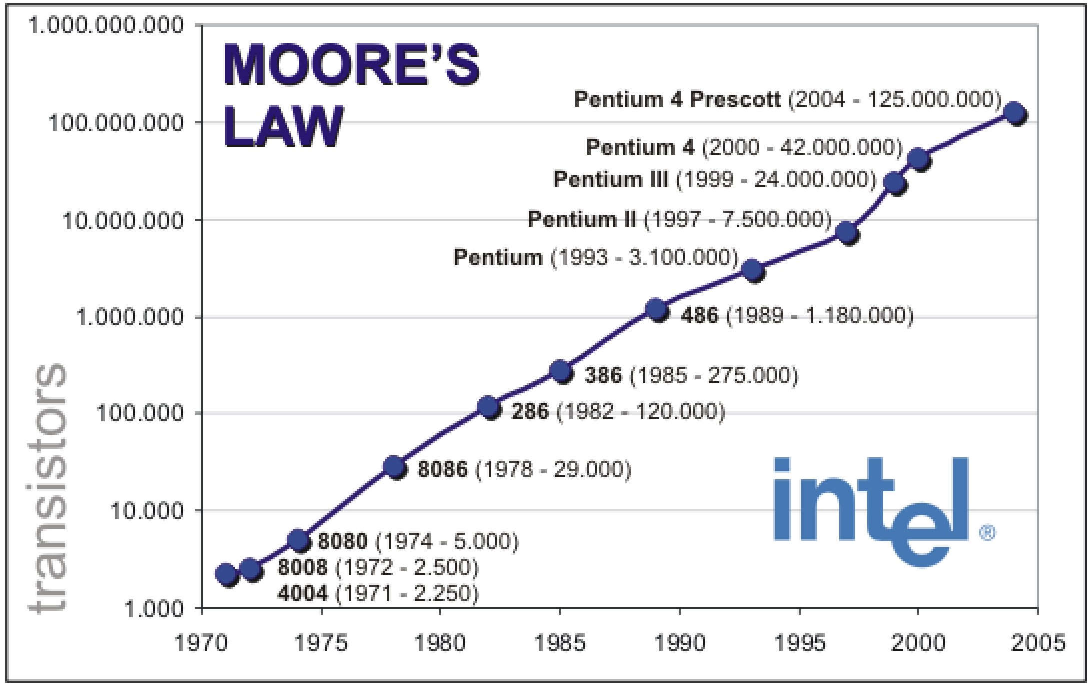
Problem nalaženja najkraće putanje koja **počinje na proizvoljnoj lokaciji, prolazi kroz svaku preostalu lokaciju tačno jednom** i **vraća se na početnu lokaciju**.

Problem trgovačkog putnika
(Traveling Salesman Problem)

→ *NP-kompletni problemi*

(nije pronađeno efikasno rešenje, ali nije ni dokazano da takvo rešenje ne postoji)

Dostižemo fizičke granice



Već smo u saturaciji Moore-ovog zakona.

Problemi disipacije i kvantne fizike nas ograničavaju.

→ *Kvantni računari (van opsega predmeta)*

→ *Paralelni algoritmi (van opsega predmeta)*

Da malo popričamo ...

1. Primer problema iz života koji zahteva sortiranje.
2. Koje strukture podataka poznajete?
3. Uporedimo problem **najkraće putanje** sa problemom **trgovačkog putnika**.
4. Primer problema iz života za koji je zadovoljavajuće dovoljno dobro rešenje.

Algoritmi su tehnologija

Šta možemo iskoristiti kao meru kvaliteta algoritma?

- **vreme** izvršavanja
- **memorijski prostor** potreban da se izvrši algoritam

Da su vreme i prostor **neograničeni**, svako **tačno** rešenje bi bilo i prihvatljivo.

Međutim, neka **tačna** rešenja su suviše **spora** da bi bila prihvatljiva. Brute Force

Zato učimo teoriju algoritama!

Kako ćemo meriti efikasnost algoritma?

Najčešće će nam najvažnija metrika biti **vreme izvršavanja algoritma** u zavisnosti od **veličine ulaza**.

Kako to merimo?

$$\boldsymbol{lg = log_2}$$

Npr. naučićemo nekoliko načina sortiranja niza elemenata:

Insertion sort **vreme izvršavanja** $\boldsymbol{t = C_1 \times N^2}$

Merge sort **vreme izvršavanja** $\boldsymbol{t = C_2 \times N \times lg(N)}$

Zašto je efikasnost toliko važna?

Računar A

$$10 \times 10^9 \; \frac{instr}{sec}$$

Insertion sort

$$t = 2n^2$$

Računar B

$$10 \times 10^6 \; \frac{instr}{sec}$$

Merge sort

$$t = 50nlg(n)$$

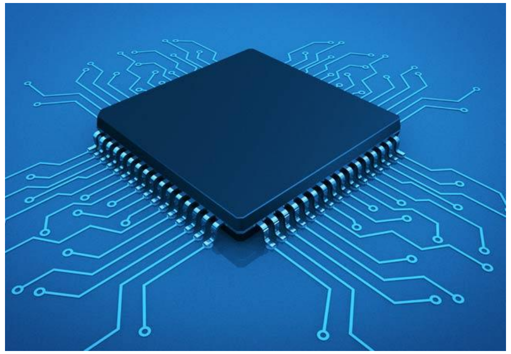
$$n = 10^7$$

$$\frac{2 \times (10^7)^2 \; instr}{10^{10} \; instr/sec} = 20{,}000 \; sec$$

$$\frac{50 \times 10^7 \times \lg(10^7) \; instr}{10^7 \; instr/sec} \approx 1{,}163 \; sec$$

$$n = 10^8 \;\; ???$$

Zašto učimo algoritme?



Algoritmi su srž većine računarskih tehnologija!

Zašto učimo algoritme?

Moderni izazovi zahtevaju obradu sve veće količine podataka.

Dobar od lošeg tačnog algoritma možemo da razlikujemo samo na **velikom broju podataka** koje obrađujemo.

Danas nam ne treba znanje algoritama ako želimo da napravimo Web stranicu ili dizajniramo GUI, ali ako imamo neka znanja iz algoritama možemo uraditi mnogo, mnogo više!

Vežbe

1. Primer aplikacije koja zahteva dobar algoritam i opis algoritma koji je potreban.

2. Ukoliko algoritam A ima vreme izvršavanja $t_A = 100n^2$, a algoritam B ima vreme izvršavanja $t_B = 2^n$, da li algoritam A ikada pobedi algoritam B i za koju veličinu ulaznih podataka?

Vežbe

1. Primer aplikacije koja zahteva dobar algoritam i opis algoritma koji je potreban.

2. Ukoliko algoritam A ima vreme izvršavanja $t_A = 100n^2$, a algoritam B ima vreme izvršavanja $t_B = 2^n$, da li algoritam A ikada pobedi algoritam B i za koju veličinu ulaznih podataka?

$$n = 15$$
$$t_A = 22{,}500$$
$$t_B = 32{,}768$$

Problem

1. Za svaku funkciju $f(n)$ iz tabele, izračunati najveću veličinu ulaza n tako da se algoritam može završiti u vremenu t. Pretpostaviti da izvršenje algoritma zahteva $f(n)$ mikrosekundi.

f(n) / t	1 sec	1 min	1 h	1 dan	1 vek
$\lg n$					
n					
n^2					
2^n					
$n!$					

Problem

1. Za svaku funkciju $f(n)$ iz tabele, izračunati najveću veličinu ulaza n tako da se algoritam može završiti u vremenu t. Pretpostaviti da izvršenje algoritma zahteva $f(n)$ mikrosekundi.

f(n) / t	1 sec	1 min	1 h	1 dan	1 vek
$\lg n$					
n	$\mathbf{10^6}$	$\mathbf{6 \times 10^7}$	$\mathbf{3.6 \times 10^9}$	$\mathbf{8.64 \times 10^{10}}$	$\mathbf{3.16 \times 10^{15}}$
n^2					
2^n					
$n!$					

Problem

1. Za svaku funkciju $f(n)$ iz tabele, izračunati najveću veličinu ulaza n tako da se algoritam može završiti u vremenu t. Pretpostaviti da izvršenje algoritma zahteva $f(n)$ mikrosekundi.

f(n) / t	1 sec	1 min	1 h	1 dan	1 vek
$\lg n$	$\mathbf{2^{10^6}}$	$\mathbf{2^{6\times10^7}}$	$\mathbf{2^{3.6\times10^9}}$	$\mathbf{2^{8.64\times10^{10}}}$	$\mathbf{2^{3.16\times10^{15}}}$
n	$\mathbf{10^6}$	$\mathbf{6\times10^7}$	$\mathbf{3.6\times10^9}$	$\mathbf{8.64\times10^{10}}$	$\mathbf{3.16\times10^{15}}$
n^2					
2^n					
$n!$					

Problem

1. Za svaku funkciju $f(n)$ iz tabele, izračunati najveću veličinu ulaza n tako da se algoritam može završiti u vremenu t. Pretpostaviti da izvršenje algoritma zahteva $f(n)$ mikrosekundi.

f(n) / t	1 sec	1 min	1 h	1 dan	1 vek
$\lg n$	$\mathbf{2^{10^6}}$	$\mathbf{2^{6\times10^7}}$	$\mathbf{2^{3.6\times10^9}}$	$\mathbf{2^{8.64\times10^{10}}}$	$\mathbf{2^{3.16\times10^{15}}}$
n	$\mathbf{10^6}$	$\mathbf{6\times10^7}$	$\mathbf{3.6\times10^9}$	$\mathbf{8.64\times10^{10}}$	$\mathbf{3.16\times10^{15}}$
n^2	**1,000**	**7,745**	**60,000**	**293,938**	$\mathbf{5.62\times10^7}$
2^n					
$n!$					

Problem

1. Za svaku funkciju $f(n)$ iz tabele, izračunati najveću veličinu ulaza n tako da se algoritam može završiti u vremenu t. Pretpostaviti da izvršenje algoritma zahteva $f(n)$ mikrosekundi.

f(n) / t	1 sec	1 min	1 h	1 dan	1 vek
$\lg n$	2^{10^6}	$2^{6\times 10^7}$	$2^{3.6\times 10^9}$	$2^{8.64\times 10^{10}}$	$2^{3.16\times 10^{15}}$
n	10^6	6×10^7	3.6×10^9	8.64×10^{10}	3.16×10^{15}
n^2	1,000	7,745	60,000	293,938	5.62×10^7
2^n	19	25	31	36	51
$n!$					

Problem

1. Za svaku funkciju $f(n)$ iz tabele, izračunati najveću veličinu ulaza n tako da se algoritam može završiti u vremenu t. Pretpostaviti da izvršenje algoritma zahteva $f(n)$ mikrosekundi.

f(n) / t	1 sec	1 min	1 h	1 dan	1 vek
$\lg n$	2^{10^6}	$2^{6\times10^7}$	$2^{3.6\times10^9}$	$2^{8.64\times10^{10}}$	$2^{3.16\times10^{15}}$
n	10^6	6×10^7	3.6×10^9	8.64×10^{10}	3.16×10^{15}
n^2	1,000	7,745	60,000	293,938	5.62×10^7
2^n	19	25	31	36	51
$n!$	9	11	12	13	17

Problem

1. Za svaku funkciju $f(n)$ iz tabele, izračunati najveću veličinu ulaza n tako da se algoritam može završiti u vremenu t. Pretpostaviti da izvršenje algoritma zahteva $f(n)$ mikrosekundi.

Svemir ima ~10^{79} atoma!

f(n) / t	1 sec	1 min	1 h	1 dan	1 vek
$\lg n$	2^{10^6}	$2^{6\times10^7}$	$2^{3.6\times10^9}$	$2^{8.64\times10^{10}}$	$2^{3.16\times10^{15}}$
n	10^6	6×10^7	3.6×10^9	8.64×10^{10}	3.16×10^{15}
n^2	1,000	7,745	60,000	293,938	5.62×10^7
2^n	19	25	31	36	51
$n!$	9	11	12	13	17


thank you!