VILNIAUS TECHNOLOGIJŲ IR VERSLO
PROFESINIO MOKYMO CENTRAS

7- ALGORITMAI

Jaroslav Grablevski / Justina Balsė

Algoritmas

- Algoritmas – tai veiksmų seka ar metodas, kuris naudojamas siekiant išspręsti užduotį ar uždavinį.
- Algoritmas gali būti suprantamas kaip funkciją, kuri apdoroja įėjimo duomenis ir gražina rezultatus.

Algoritmas

- Algoritmai užrašomas:
  - Schemomis (srautų diagrama)
  - Abstrakčia kalba (pseudokodu)
  - Normalia kalba
- Algoritmas realizuojamas viena iš programavimo kalbų

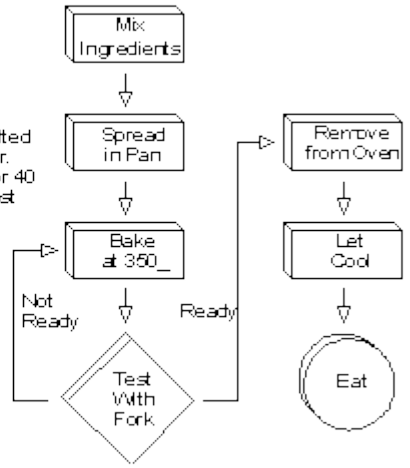
Recipe
CHOCOLATE CAKE

4 oz. chocolate
1 cup butter
2 cups sugar
3 eggs
1 tsp. vanilla
1 cup flour

Melt chocolate and butter. Stir sugar into melted chocolate. Stir in eggs and vanilla. Mix in flour. Spread mix in greased pan. Bake at 350_ for 40 minutes or until inserted fork comes out almost clean. Cool in pan before eating.

Program Code

```
Declare variables:
    chocolate   eggs        mix
    butter      vanilla
    sugar       flour
mix = melted ((4*chocolate) + butter)
mix = stir (mix + (2*sugar))
mix = stir (mix + (3*eggs) + vanilla)
mix = mix + flour
spread (mix)
While not clean (fork)
bake (mix, 350)
```



Pirminio skaičiaus nustatymas

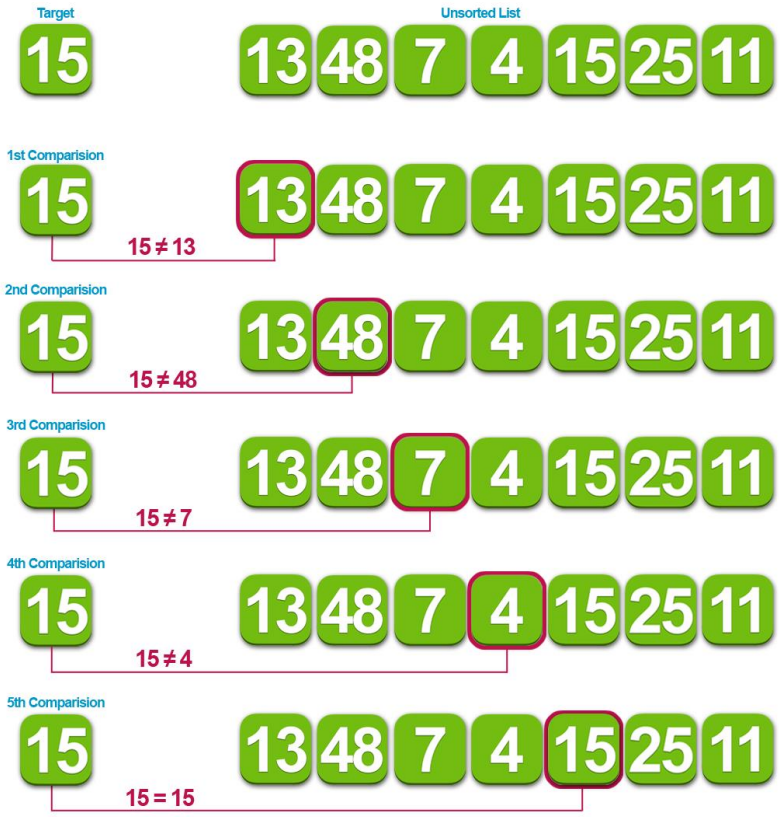
```
public static boolean arPirminis(long skaicius) {
    for (long i = 2; i < skaicius - 1; i++) {
        if (skaicius % i == 0)
            return false;
    }
    return true;
}
```

Pirminio skaičiaus nustatymas

```
public static boolean arPirminis2(long skaicius) {
    if (skaicius % 2 == 0)
        return false;
    long max = (long) Math.floor(Math.sqrt(skaicius));
    for (int j = 3; j <= max; j += 2) {
        if (skaicius % j == 0)
            return false;
    }
    return true;
}
```

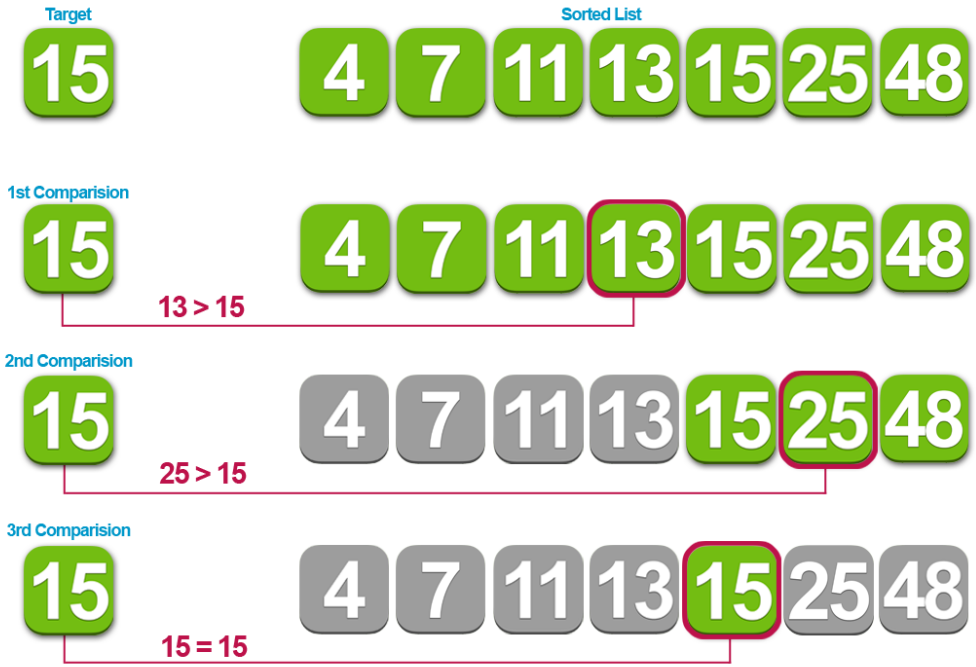
Nuosekli paieška (linear or sequential search)

- Primityviausias reikiamo rakto paieškos masyve būdas yra pradėti nuo masyvo pradžios ir peržiūrėti iš eilės kiekvieną elementą



Dvejetainė paieška (binary search)

- **Dvejetainė paieška** yra paieškos sutvarkytame sąraše procesas. Jis dalina sutvarkytą sąrašą tol, kol atsiduria vietoje, kurioje yra ieškomas elementas, jei jis iš viso sąraše yra.

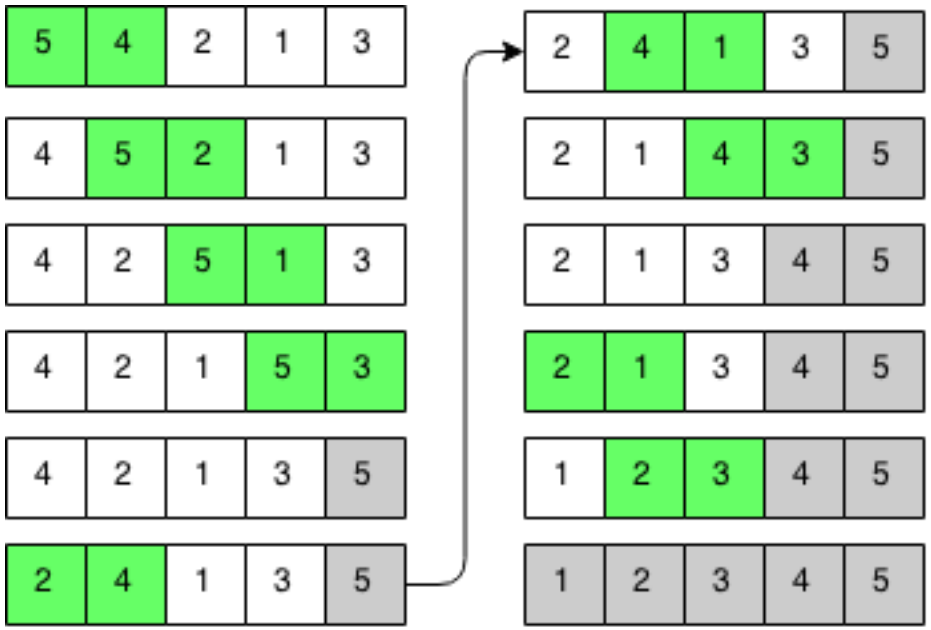


Rikiavimo algoritmai.

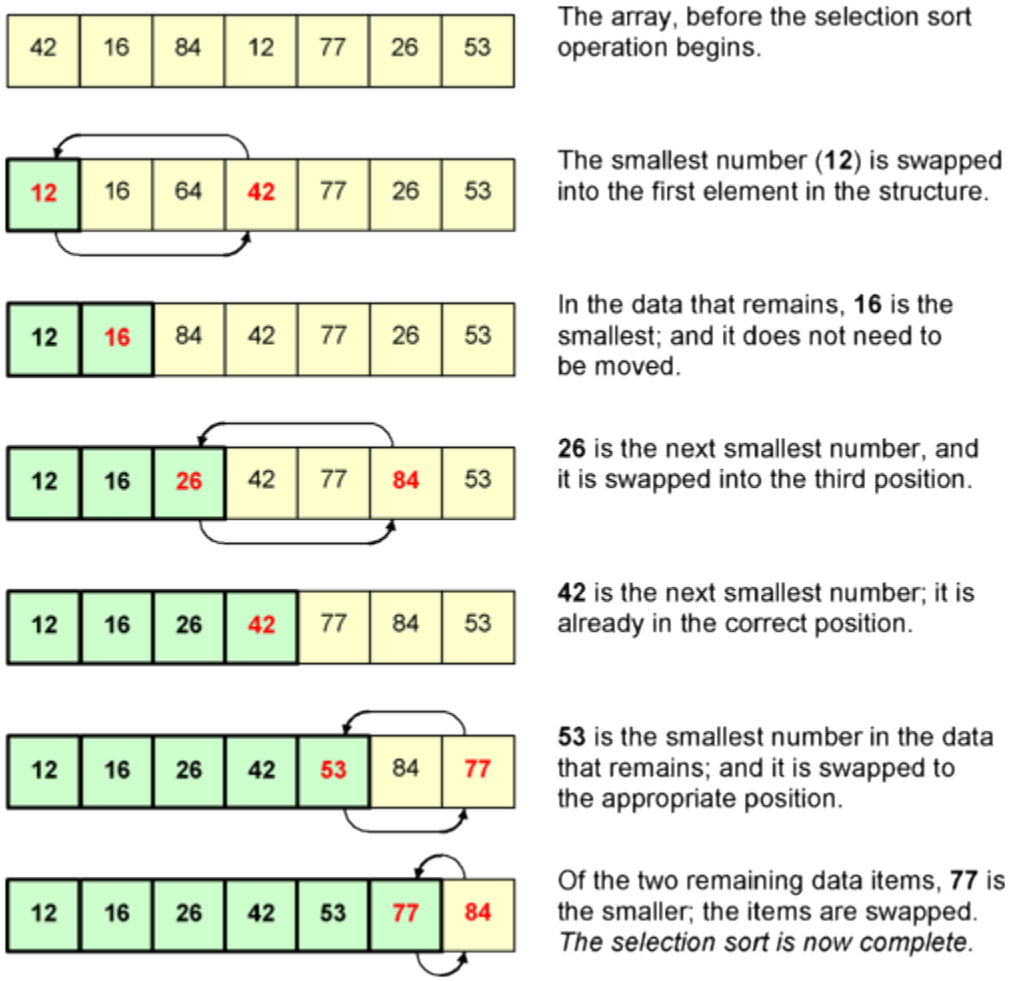
- **Rikiavimo algoritmas** – algoritmas, dėstantis duomenis tam tikra tvarka.
- Rikiavimo algoritmai gali būti skirstomi keliais būdais:
  - Pagal sudėtingumą.
  - Pagal naudojamą atmintį.
  - Pagal stabilumą.
- Daugiau informacijos:
  - http://bfy.tw/20gG
  - https://www.toptal.com/developers/sorting-algorithms
  - https://visualgo.net/bn/sorting?slide=1
  - Java Algorithms (Derek Banas)

Bubble sort

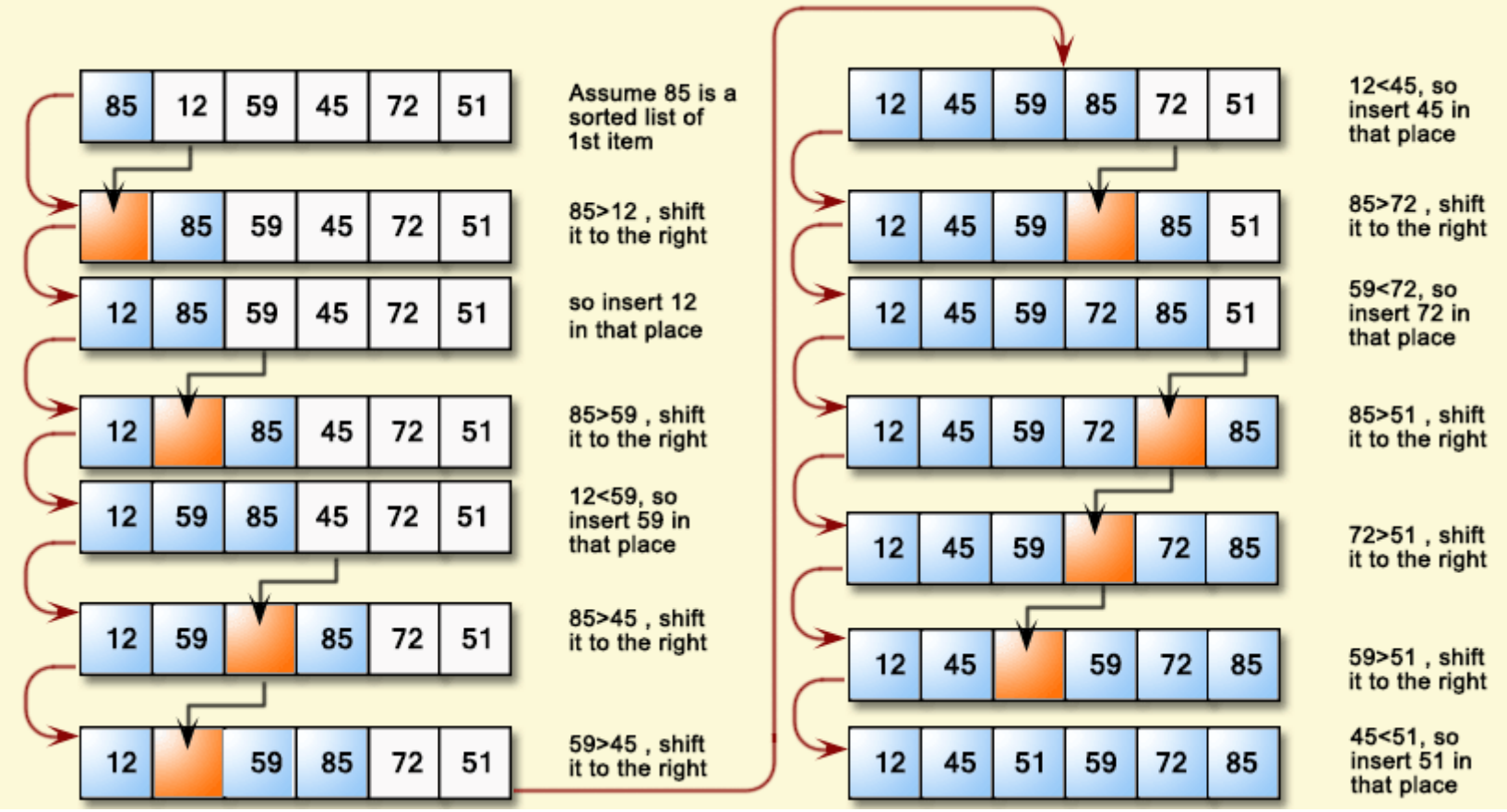
- Pagrindinė algoritmo idėja yra lyginti gretimus elementus ir, jeigu reikia, keisti juos vietomis. Kiekvienos iteracijos metu ne tik didžiausias iš nesurūšiuotų elementų padedamas į savo vietą, bet ir atliekamas kitų elementų patvarkymas.



Rūšiavimas paieška (*selection sort*)

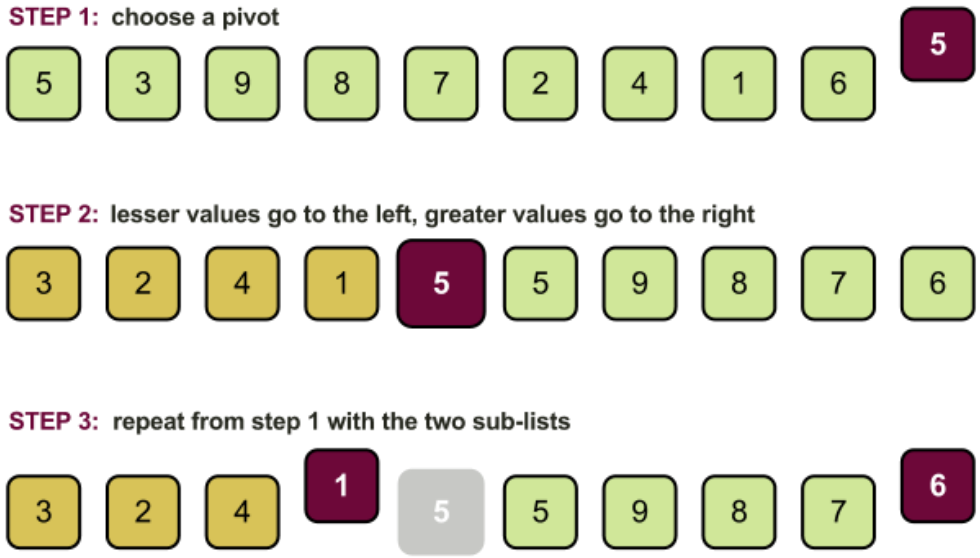


Rūšiavimas įterpimu (*insertion sort*)

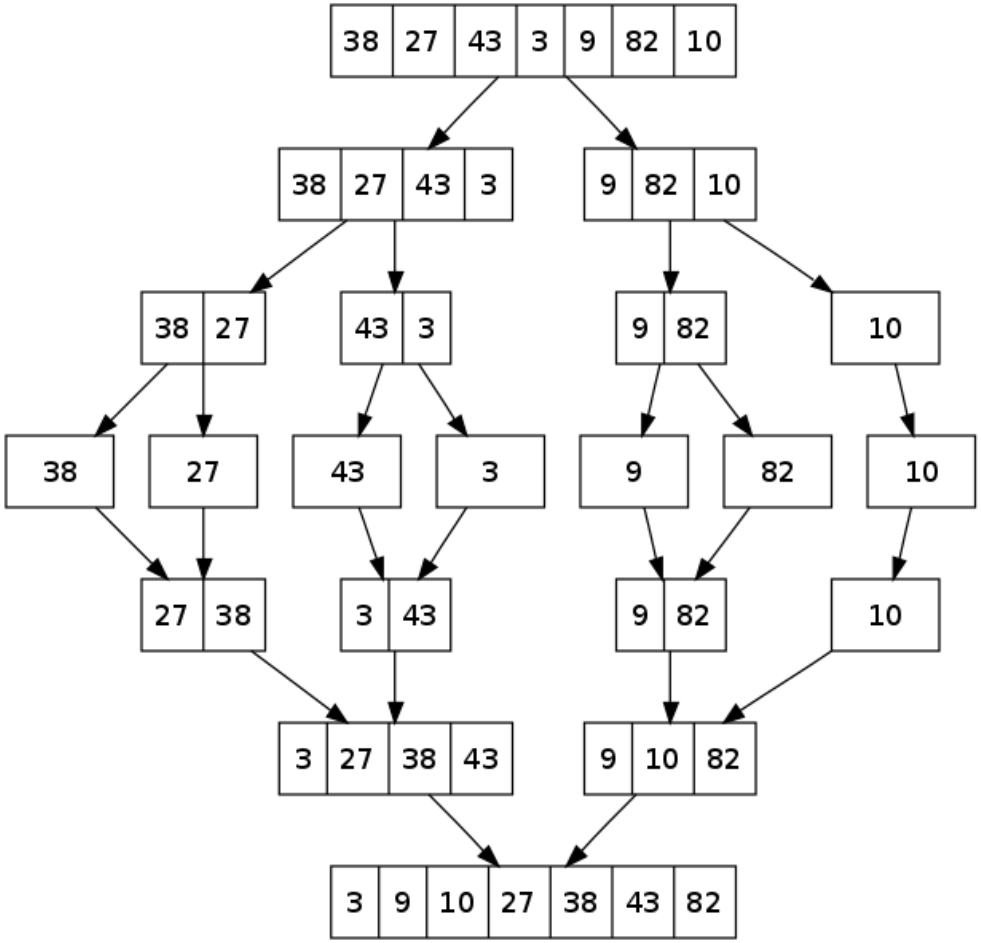


Greitasis rikiavimas (*quicksort*)

- Pasirinkti vieną masyvo elementą (bet kurį), šis elementas vadinamas „pivot“ elementu;
  2. Likusius elementus suskirstyti į dvi grupes: a) kurie yra mažesni už pivot elementą, b) kurie yra didesni už pivot elementą;
  3. Rekursyviai rūšiuoti kiekvieną grupę atskirai.

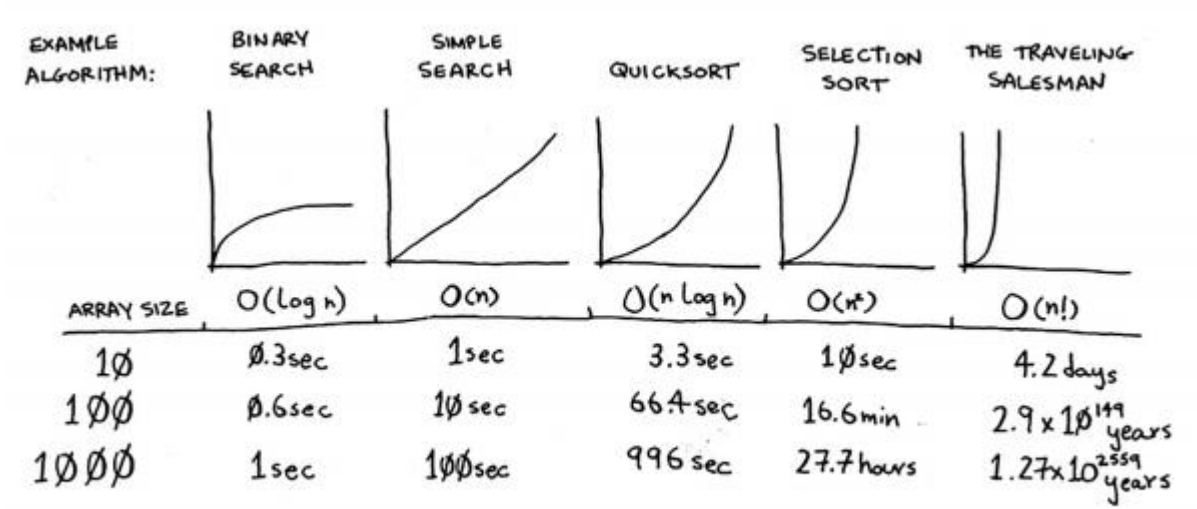


Rikiavimas sujungimu (*merge sort*)



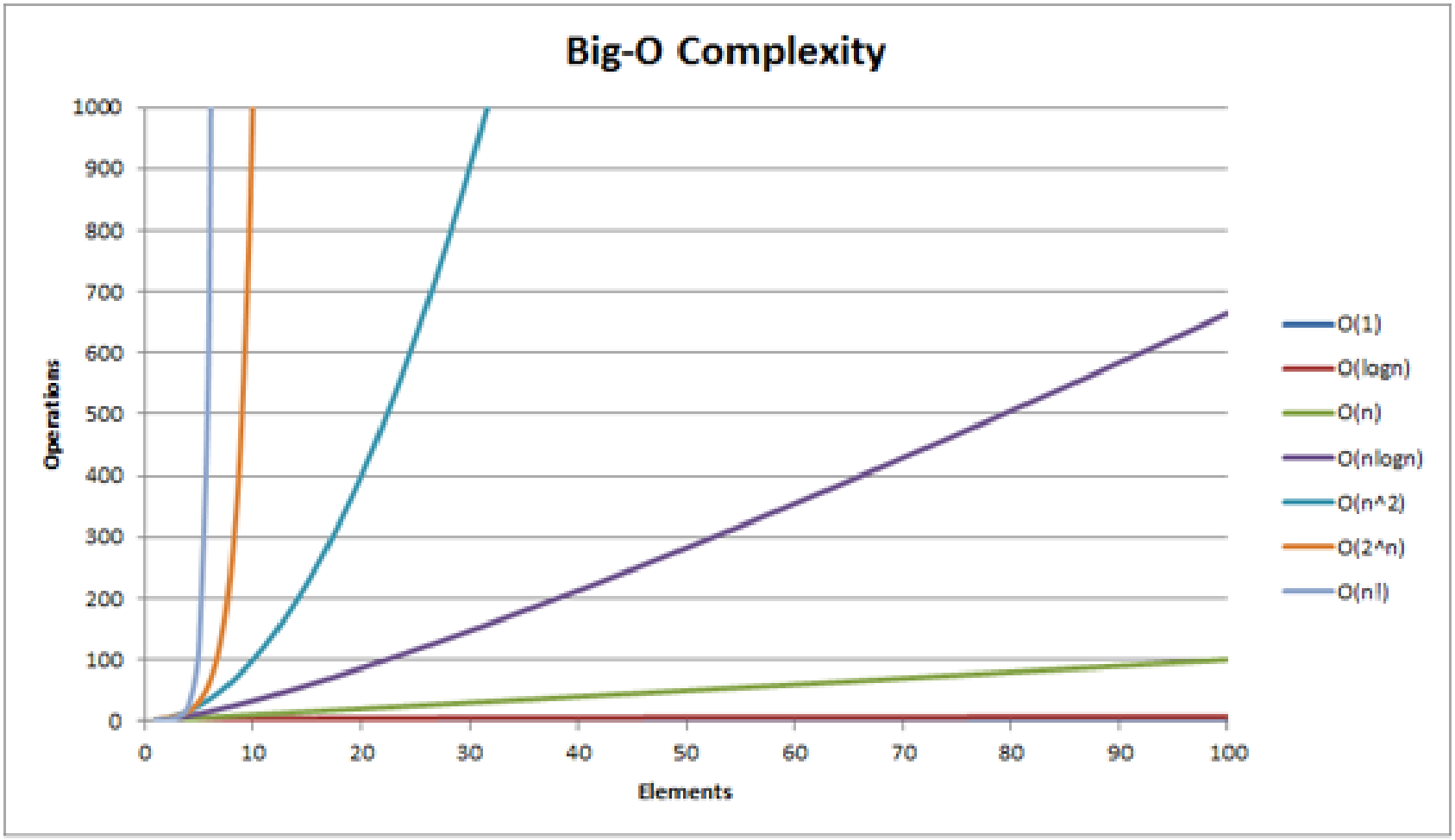
Algoritmų laiko sudėtingumas

- Laiko sudėtingumo skaičiavimas vertina, kiek laiko reiktų tam tikrai problemai su tam tikru duomenų dydžiu spręsti efektyviausiu algoritmu.
- Tarkime, kad turint n bitų duomenų kiekį, problema išsprendžiama per n^2 žingsnių; tokia problema yra n^2 sudėtingumo.



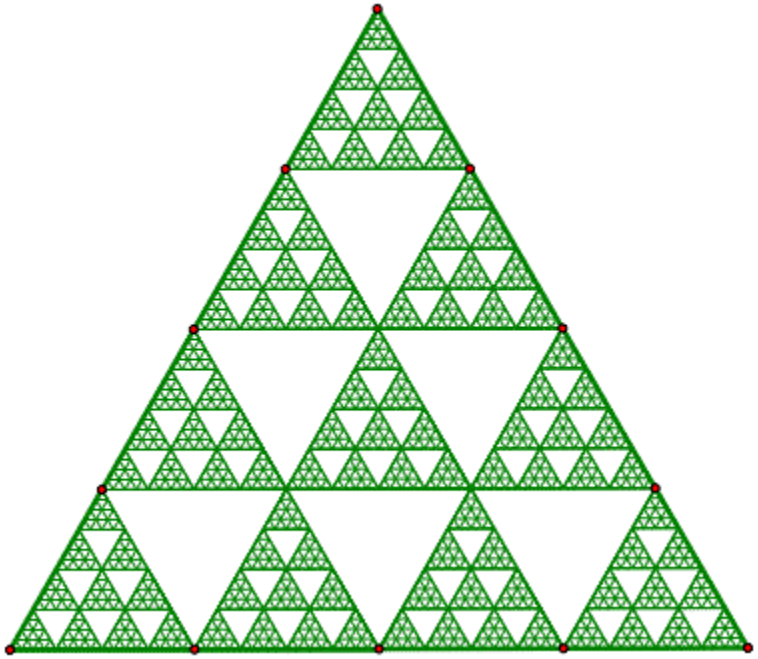
Estimates based on a slow computer that performs 10 operations per second

Big O notation



Rekursija

- Programų ar algoritmų sudarymo metodas, kai programa kreipiasi pati į save, esant mažesnėms argumentų reikšmėms.

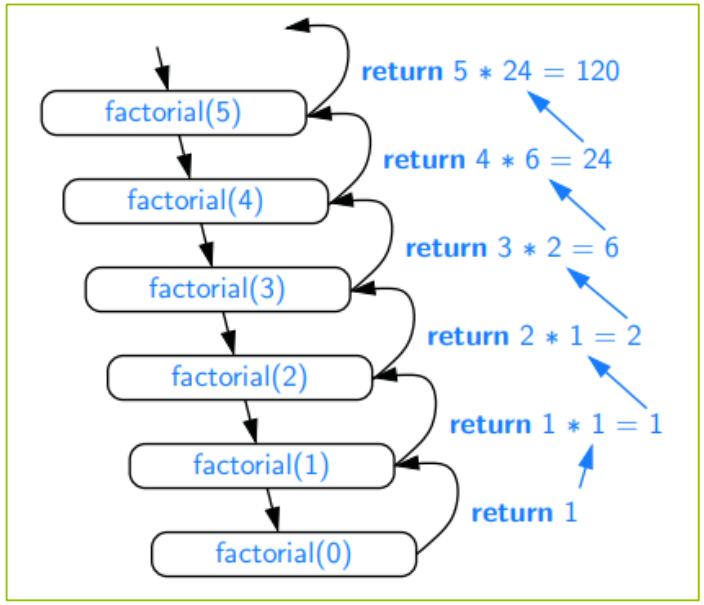


Rekursija

- Algoritmas vadinamas rekursiniu, jei jis kviečia save patį atlikti dalį skaičiavimų
- Rekursinis algoritmas privalo turėti dvi dalis:
  - Bazinę dalį, kuri sprendžiama nenaudojant rekursijos;
  - Rekursinę dalį, kurioje atliekamas tos pačios funkcijos kvietimas.
- Kiekvienoje rekursinėje procedūroje turi būti numatyti visi ribiniai atvejai, kuriuos pasiekus rekursija nutraukiama.
- **Rekursijos gylis –**tai ilgiausia procedūrų grandinė, kuri suskaičiuoja funkcijos reikšmę.

Rekursija (faktorialas)

```
public static long factorial(long number) {
    if (number <= 1) // test for base case
        return 1; // base cases: 0! = 1 and 1! = 1
    else // recursion step
        return number * factorial(number - 1);
}
```



Rekursija

- Rekursijos privalumai
  - Patogiai kontroliuojama procedūros eiga
  - Paprastas kodas
- Rekursijos trūkumai
  - Stack overflow
  - Nevisada efektyvus algoritmas
- Rekomendacija
  - Nenaudoti rekursijos, jei nežinomas elementų skaičius.