

# DUOMENŲ STRUKTŪROS IR ALGORITMAI

---

**Algoritmai:** tipai, charakteristikos, struktūrinė  
schema

# Praeitios paskaitos santrauka

- AVL medis
  - Savybės, balansinis faktorius
  - Viršūnių pridėjimas ir šalinimas
  - Posūkiai į kairę ir į dešinę
- Bajero medis
  - Daugiašakis medis, medžio eilė, viršūnės raktai
  - Formavimas, rakto pridėjimas, šalinimas
- Piramidė
  - Savybės
  - Realizacija masyve

# Algoritmai

**Algoritmas - tai tam tikra veiksmų seka, kurią reikia atlikti norint pasiekti rezultatą.**

*Algoritmo* sinonimai: instrukcijos, nurodymai, taisyklės.

Bendrai algoritmą būtų galima apibūdinti kaip **tikslių nurodymų seką tam, kas turės atlikti konkrečią užduotį.**

Daugelį kasdieninės veiklos rezultatų pasiekiamo naudodami algoritmus t.y. tam tikrą veiksmų seką.

Kartais sukeitus algoritmo veiksmus, rezultatas nepakinta. Tačiau vykdant kai kuriuos algoritmus veiksmų sukeitimas gali sugriauti visą tolimesnę algoritmo eigą.

# Algoritmai

## Algoritmas turi patenkinti šias sąlygas:

1. Atlikti konkretų veiksma ir pateikti rezultata;
2. Būti aiškus ir nedviprasmiškas;
3. Apibrėžti žingsnių seką ir jų atlikimo tvarką, reikalingą rezultatui gauti.
4. Būti baigtinis atliekamų žingsnių skaičiumi
5. Būti baigtinis laiko ir resursų atžvilgiu t. y. kiekvienas žingsnis turi būti toks, kad jį būtų galima atlikti.

Algoritmai, tenkinantys 1-3 sąlygas vadinami daliniais (angl. partial) algoritmais.

Algoritmai tenkinantys visas penkias sąlygas – pilnais (angl. total) algoritmais.

# Algoritmų savybės

Algoritmams būdingos tokios bendrosios savybės:

- **Diskretumas** (algoritmas skaidomas į tiksliai aprašytus vykdymo žingsnius).
- **Baigtumas** (algoritmas turi turėti pabaigą).
- **Rezultatyvumas** (algoritmas visada turi pateikti konkretų rezultatą, jei jis egzistuoja).
- **Aiškumas** (algoritmas turi būti pateikiamas taip, kad jį visi vienareikšmiškai suprastų).
- **Universalumas** (algoritmas turi tikti bet kokiems duomenimis).

# Algoritmų efektyvumas

Algoritmo efektyvumas yra apibrėžiamas pagal tokius kriterijus:

- Atliekamų bazinių operacijų skaičių
- Atminties kiekį, kuris reikalingas algoritmui realizuoti.

## Algoritmų tipai:

- Tiesiniai (  $y = a \cdot x + b$  )
- Šakotieji (  $y = a \cdot x + b$  (if  $x < 0$ ) arba  $y = 2x - 1$  (if  $x > 0$ ) )
- Sudėtingi (pvz. e-laiško siuntimas)

# Pavyzdys

Sakykime, kad reikia sudauginti dvi kvadratinės matricas:  
A [n, n] ir B [n, n].

Rezultatų matricos  $C = A * B$  elementai bus randami pagal tokią formulę:

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}, \quad 1 \leq i, j \leq n.$$

Sandaugos veiksmų skaičius:  $n^3$

Sudėties veiksmų skaičius:  $n^2(n - 1)$

Bendras veiksmų skaičius  $2n^3 - n^2$ .

# Tiesiniai algoritmai

**Tiesiniais algoritmais** vadinami tokie algoritmai, kuriuose veiksmai atliekami nuosekliai vienas po kito be jokių alternatyvų ar veiksmų kartojimo.

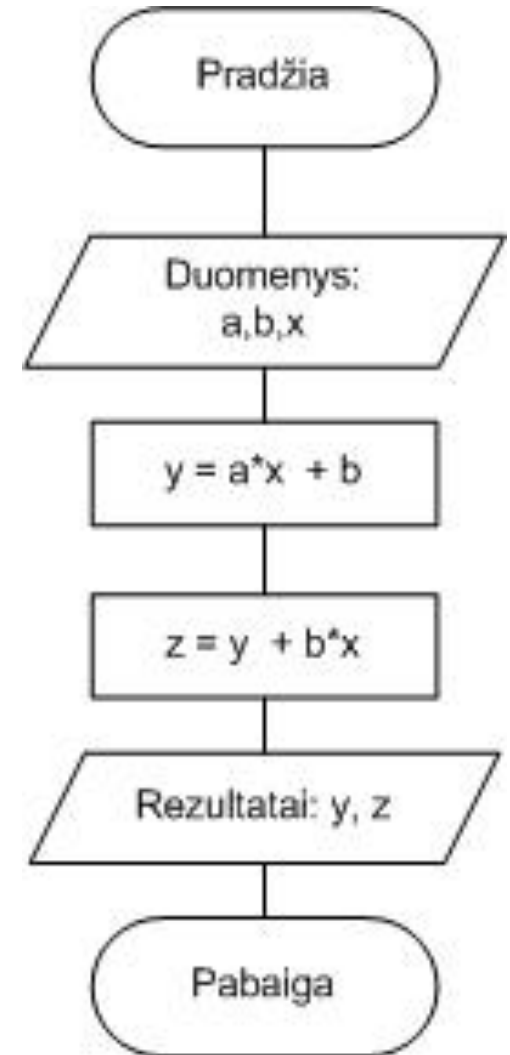
Tokie algoritmai naudojami dideliuose su sudėtinguose algoritmuose, kaip jų dalis.

## Pavyzdys

Apskaičiuoti funkcijų reikšmes:

$$y = a * x + b$$

$$z = y + b * x$$





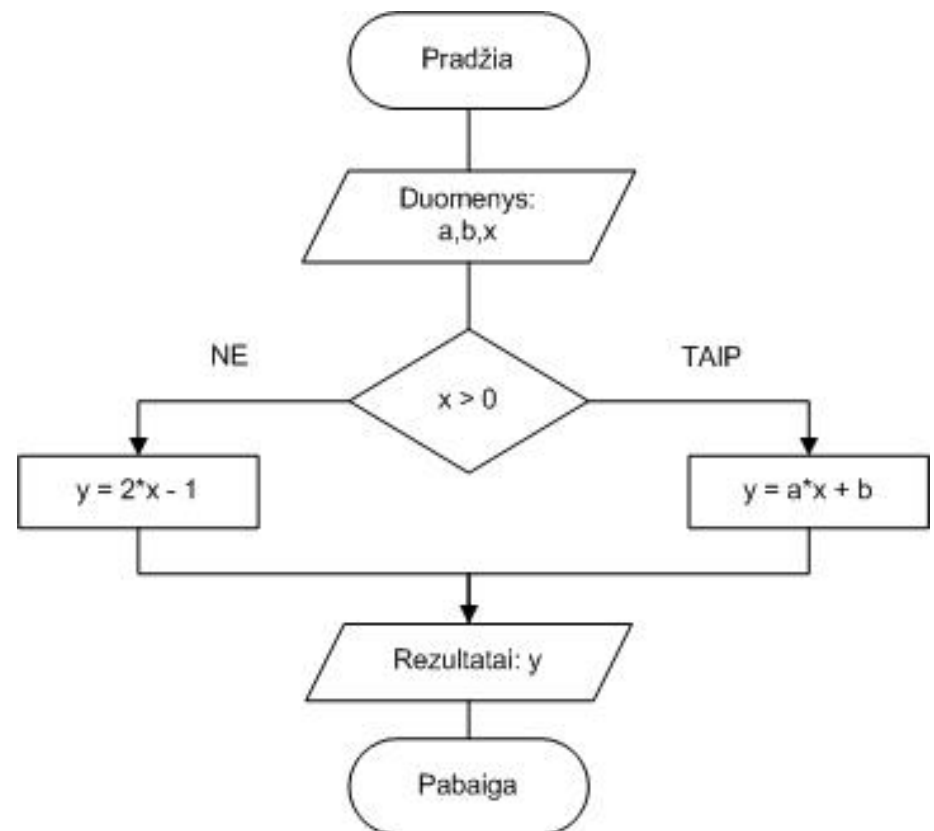
# Šakotieji algoritmai

**Šakotaisiais algoritmais** vadinami tokie algoritmai, kuriuose yra alternatyvūs sprendimo keliai, parenkami priklausomai nuo tikrinamos sąlygos rezultato.

## Pavyzdys

Rasti funkcijos reikšmes

$$y = \begin{cases} a * x + b, & x > 0 \\ 2 * x - 1, & x < 0 \end{cases}$$



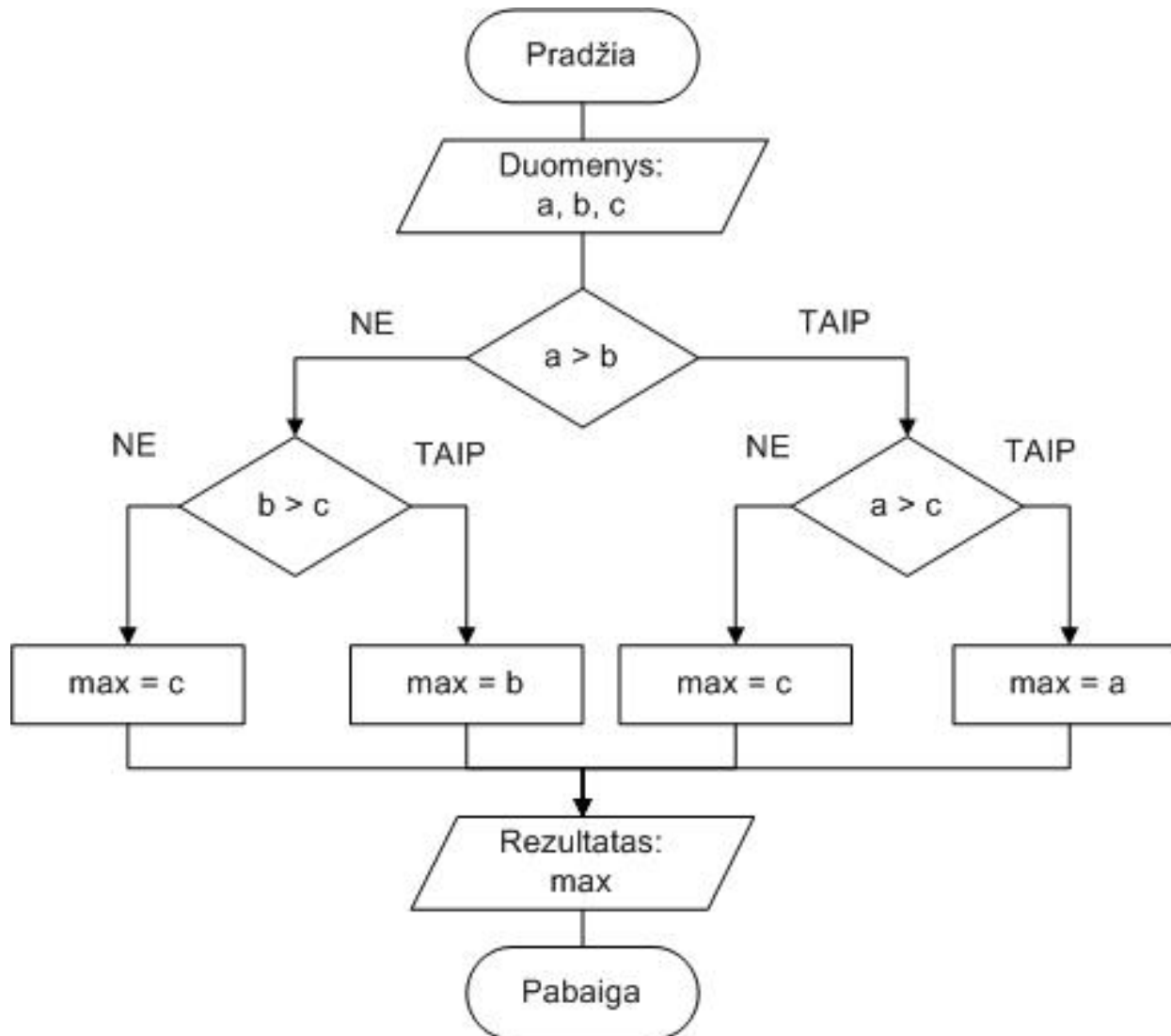
# Sudėtingi algoritmai

**Sudėtingi algoritmai** - tai algoritmai, apjungiantys tiesinius ir šakotuosius algoritmus. Būtent tokio tipo algoritmai naudojami realių uždavinių sprendimui.

## Pavyzdys

Surasti didžiausią skaičių.

# Sudėtingi algoritmai



# Algoritmų vaizdavimas

Algoritmai gali būti vaizduojami:

- Diagramomis
  - Pseudo programiniu kodu
  - Aprašymu
- 

Faktorialo  
skaičiavimo  
algoritmo pseudo  
kodas



```
begin int Factorial (n)  
    s = 1;  
    for ( i = 2; i <= n ; i++ )  
        s = s * i;  
    return ( s );  
end Factorial
```

# Struktūrinė schema

Struktūrinė schema naudojama siekiant vizualiai pavaizduoti sudėtingą procesą ar algoritmą.

Jos padeda skaitytojui lengviau suprasti algoritmą bei rasti problemines sritis.

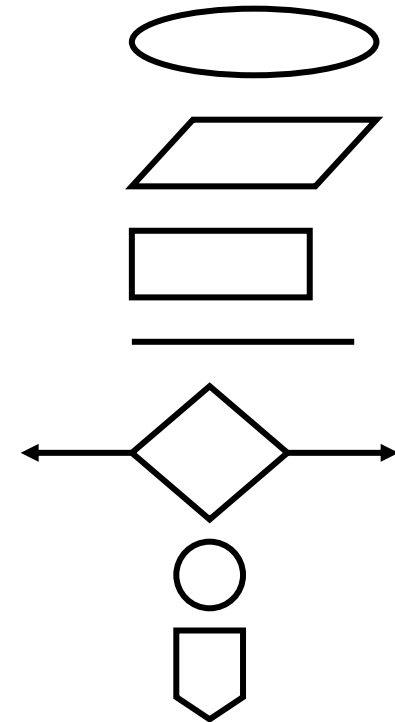
Schemoms braižyti naudojamos įvairios geometrinės figūros. Dažniausiai naudojamas:

- **Stačiakampis** - tai figūra, reiškianti skaičiavimo ar veiklos žingsnį.
- **Rombas** – tai figūra naudojama sprendimo žingsniui žymėti.

# Struktūrinė schema

## Struktūrinės schemas simboliai:

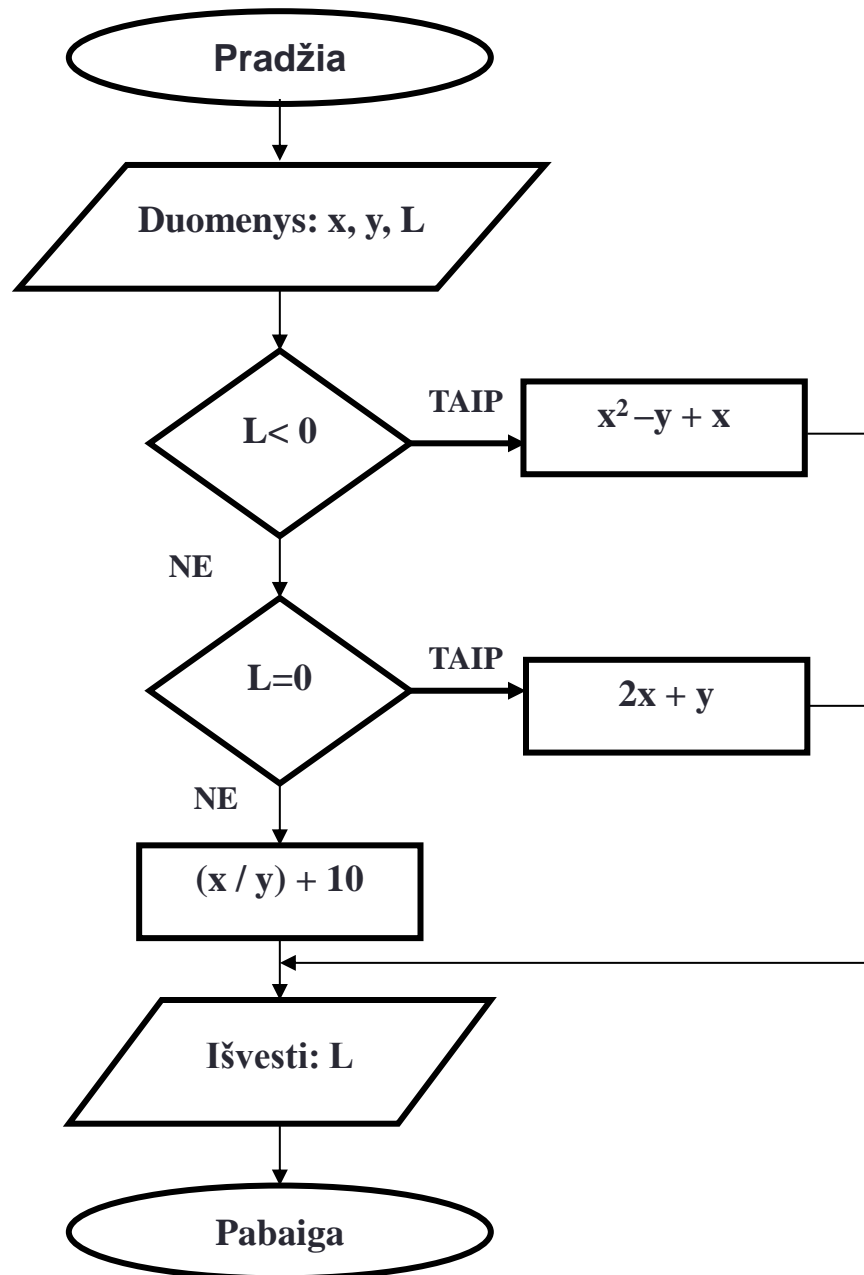
1. Terminalas (start, stop)
2. I/O duomenys
3. Apdorojimas
4. Srautas: susijęs su kryptimi
5. Sprendimas atliekant loginį veiksmą
6. Jungtis
7. Sujungimas



# Pavyzdys

Apskaičiuoti funkcijos  $F$  reikšmę:

$$F = \begin{cases} x^2 - y + x, & \text{kai } L < 0 \\ 2x + y, & \text{kai } L = 0 \\ \frac{x}{y} + 10, & \text{kai } L > 0 \end{cases}$$



# Pavyzdys

## Faktorialo skaičiavimas

```
int Factorial (N)
{ F = 1;
  for ( M = 1; M <= N ; M++ )
    F = F * M;
  return F;
```

