

DUOMENŲ STRUKTŪROS IR ALGORITMAI

Duomenys, duomenų struktūros, abstraktūs duomenų tipai, masyvai

Dėstytojas

VGTU Informacinių sistemų katedros vedėjas

prof. dr. Dalius Mažeika

e-mail Dalius.Mazeika@vgtu.lt

Tel. 27449830

<http://dma.vgtu.lt>

Paskaitos

- Teorija – antradieniais ... penktadieniais 8:00
- Pratybos – trečiadieniais... penktadieniais 9:45

Galutinio pažymio struktūra

- Egzaminas – koeficientas 0.6
- Pratybos – koeficientas 0.4
 - 2 x namų darbai po 0.15
 - 1 x namų darbas po 0.1

Prie egzamino prileidžiami atsiskaitę už 3 namų darbus ir visas klasėje darytas užduotis.

Literatūra

- R.Čiegis. **Duomenų struktūros, algoritmai ir jų analizė.** Vilnius, Technika 2007.
- Baniulis, Tamulynas. **Duomenų struktūros.** Kaunas, Technologija, 2003.
- A.Juozapavičius. **Duomenų struktūros ir algoritmai.** Vilnius, VU. 1997.
- T.Cormen, etc. **Introduction to Algorithms.** MIT, 2009.
- W.Collins. **Data structures and standart template library.** McGraw Hill, 2003.
- Clifford A. Shaffer. **A Practical Introduction to Data Structures and Algorithm Analysis.** Virginia Tech Blacksburg, VA 24061, 2011.
- <http://www.academictutorials.com/data-structure/>
- <http://cprogramminglanguage.net/>
- www.techmat.vgtu.lt/konspektai

Paskaitos turinys

- Kas yra duomenys?
- Duomenų struktūros
- Algoritmai
- Ryšys tarp algoritmų ir duomenų struktūrų
- Abstraktūs duomenų tipai
- Masyvai

Įvadas

Pirminė kompiuterio programų paskirtis – ne tik atlikti skaičiavimus, bet taip pat išsaugoti ir pateikti informacija kaip galima greičiau.

The study of data structures and the algorithms that manipulate them is at the heart of computer science.

Kurso tikslas – padėti suprasti, kaip duomenų struktūros gali būti efektyviai naudojamos duomenų saugojimui, o algoritmai gali įtakoti kompiuterio atliekamų veiksmų našumą.

Kas yra duomenys?

Duomenys - sąvoka siejama su tuo kas duota, kuo remiamasi daromos išvadas arba priimami sprendimai.

- Programose duomenys – simboliai, skaičiai, masyvai, tekstai, lentelės, paveikslėliai, garsai.
- Kiekvienos programos veikimo rezultatas – duomenys.
- Duomenys turi tipus: sveiki skaičiai, realieji skaičiai, simboliai, tekstinės eilutės.
- Galima atlikti transformaciją tarp duomenų tipų.

Duomenų struktūros

Duomenų struktūros suprantamos kaip organizuota (struktūrizuota) atskirų elementų ar objektų visuma (aibė) ir jų sąveikos operacijos tarp duomenų.

Netgi sveiki arba realieji skaičiais, saugomi kompiuterio atmintyje, gali būti traktuojami kaip paprastos duomenų struktūros.

Surūšiuotas kintamųjų sąrašas – tipinis struktūrizuotos visumos pavyzdys.

Duomenų struktūra - tai duomenų aibė, kuriai apibrėžtos tam tikros operacijos ir loginiai ryšiai tarp jų.

Abstraktūs duomenų tipai

Tipas – tai duomenų reikšmės tipas.

Pavyzdys:

- Loginis tipas turi reikšmes: true / false.
- Sveikas duomenų tipas: 1, 5, 8, 12 ...

Duomenų tipas – tai reikšmės tipas kartu su operacijomis, kurios atliekamos su duomenimis.

Pavyzdys:

- Veiksmai su sveiko tipo duomenimis
 - + ; - ; / ; * ; % .

Abstraktūs duomenų tipai

Kalbant apie abstraktų duomenų tipą, paprastai atsiribojama nuo realizacijos, jos ypatumų, pabrėžiami tik:

- duomenys (tam tikra tvarka dėstomi)
- algoritmai (tam tikra tvarka atliekamos operacijos).

Pagrindinė ADT sąvokos įvedimo priežastis - tai noras abstrahuotis nuo konkrečios realizacijos.

ADT neapibrėžia konkrečių duomenų tipų realizacijų. Realizacija siejama su programavimo kalba.

Duomenų struktūros, kaip ADT

Duomenų struktūra – ADT implementacija.

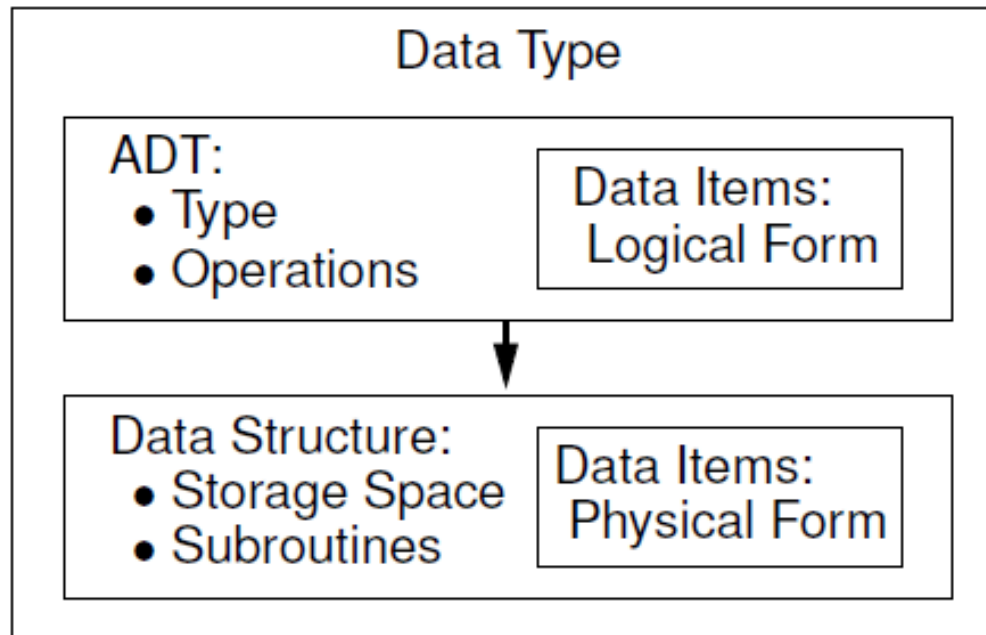
Pavyzdys:

- klasė -> objektas
- C++ (int, short int, float, double, long double, char, bool).

Sąvoka “duomenų struktūra” dažnai siejama su duomenimis, saugomais operatyviojoje atmintyje.

Panaši sąvoka **failų struktūra** siejama duomenų saugojimo organizacija periferinėse laikmenose.

ADT ... DS



ADT apibrēžia loģinē duomenų formą. Tuo tarpu duomenų struktūra apibrēžia fizinē formą.

Algoritmas

Algoritmas – tai veiksmų seka ar metodas, kuris naudojamas siekiant išspręsti užduotį ar uždavinį.

Algoritmas gali būti suprantamas kaip funkcija, kuri apdoroja įėjimo duomenis ir gražina rezultatus.



Algoritmas

Algoritmai užrašomas:

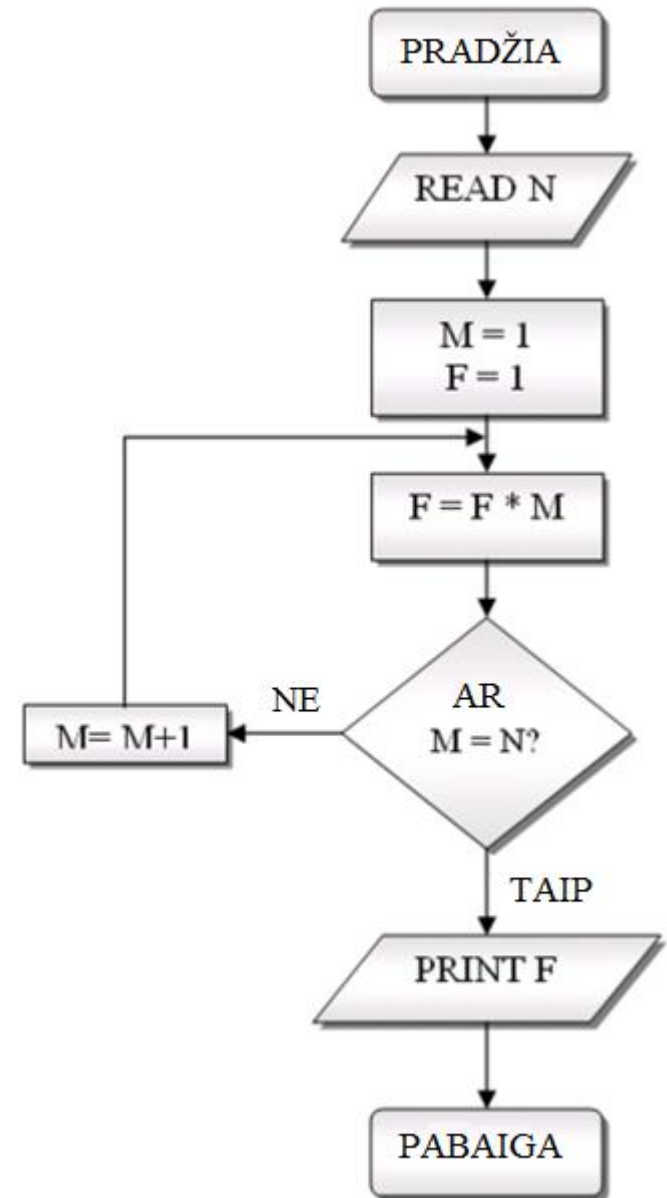
- Schemomis (srautų diagrama)
- Abstrakčia kalba
- Normalia kalba

Algoritmas realizuojamas viena iš programavimo kalbų.

Algoritmas

Faktorialo skaičiavimo algoritmo pseudokodo ir diagramos pavyzdys

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



Algoritmų savybės

Algoritmų pagrindinės savybės:

- Korektiškumas
- Baigtumas
- Apibrėžtumas
- Efektyvumas
- Algoritmas turi susidėti iš nedidelių žingsnių.
- Jis turi neužsiciklinti (**NO** *infinite loop*).

Masyvai

Masyvas – tai duomenų struktūra, pasižyminti tokiomis savybėmis:

- Visi masyvo elementai yra to paties tipo;
- Masyvo elementų skaičius yra fiksuotas ir jo negalima keisti;
- Kiekvienas masyvo elementas turi savo indeksą;
- Masyvo elementus išrenkame tiesiogiai pagal jų indeksus.

Masyvai gali būti:

- Vienmačiai
- Dvimačiai
- N-mačiai

Masyvai

Masyvo dimensijų skaičius yra tik elemento adreso vaizdavimo funkcija, leidžianti sukurti virtualų daugiamatį objektą.

Dvimatį masyvą galime vaizduoti kaip vienmačių masyvų masyvą t.y. masyvą, kurio elementai kiti vienamčiai masyvai.

Privalumas – racionaliau naudojama atmintis, nes vienmačiai masyvai gali užimti skirtingus atminties sektorius.

Masyvai

Masyvo elementai apibrėžiami naudojant indeksus:

- Vienmatis $A[0]; A[1]; A[2]$
- Dvimatis $B[0][0]; B[0][1]; B[1][2];$
- Trimatis $C[0][0][0]; C[0][0][1]; C[1][2][1];$

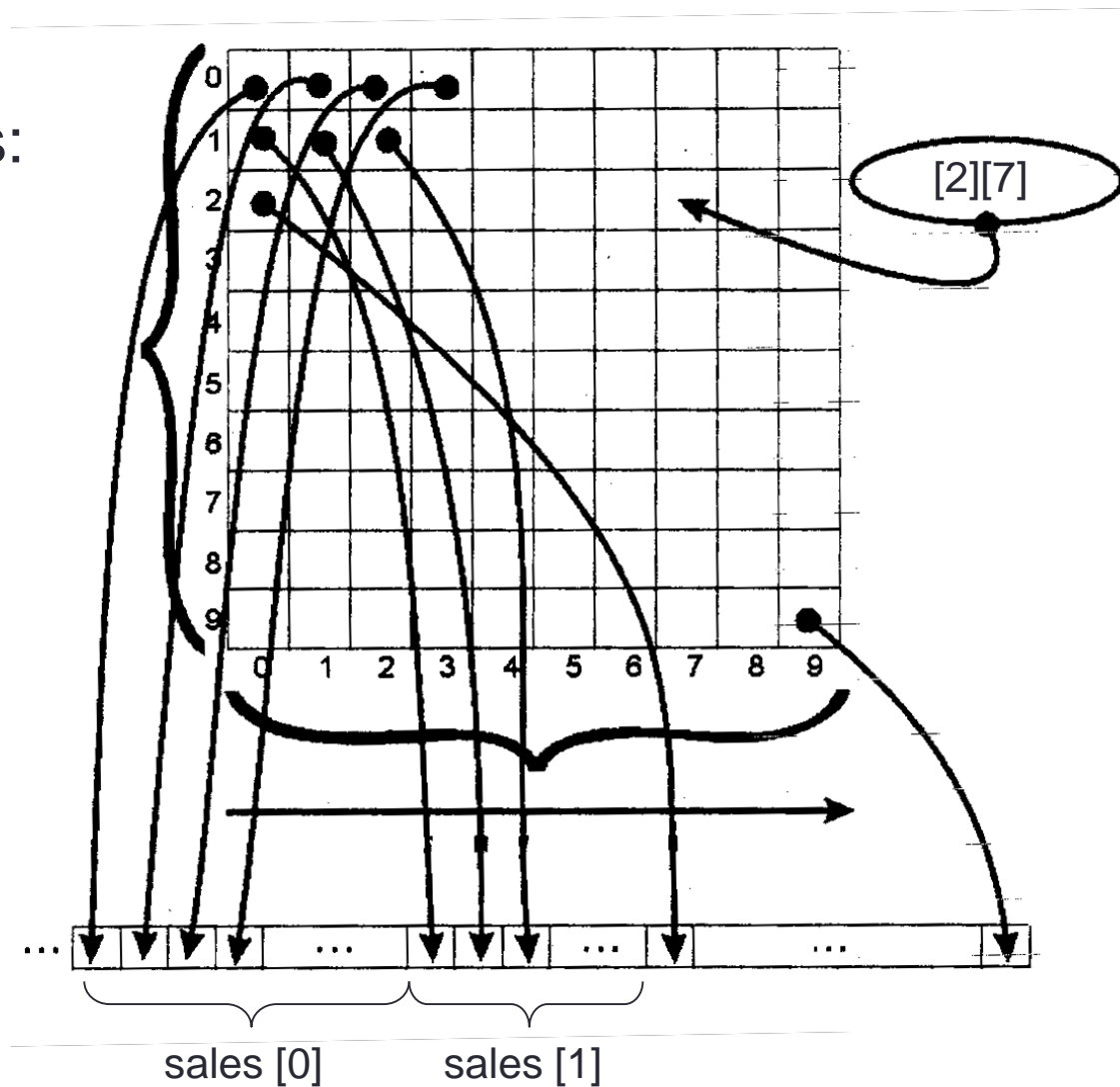
Indeksai – sveiki skaičiai. Tarkime vienmačio masyvo A indeksai priklauso intervalui $[L;U]$. Tokio masyvo elementų skaičius apskaičiuojamas taip:

$$U - L + 1$$

Dvimačiai masyvai

Dvimatis masyvas:

`sales[10][10]`



Masyvo elementų adresai

Sakykim:

- Vieno elemento ilgis lygus **S**;
- Pirmojo elemento pirmojo baido adresas **B**.

Tuomet, bet kurio vienmačio masyvo elemento $A[i]$ adresas:

$$A[i] = B + (i - L) * S$$

Dvimačio masyvo bet kurio elemento $A[i][j]$ adresas:

$$A[i][j] = B + (i - L_1)(U_2 - L_2 + 1) * S + (j - L_2) * S$$

Pratybos

Microsoft Visual C++ 2010 Express

Kvadratinės lygties sprendimas (prisiminimui).

Pirminio skaičiaus paieška:

- Primityvus algoritmas
 - Statiniame masyve
- Optimizuotas algoritmas
 - Nelyginiai skaičiai
 - Sumažinta dalybos elementų skaičius