Počítače a algoritimizácia

Aplikácie údajových typov a štruktúr pri tvorbe algoritmov

Prednášajúci:
doc. Ing. Anna Jadlovská, PhD.
doc. Ing. Ján Jadlovský, CSc.

Aplikácie údajových typov a štruktúr pri tvorbe algoritmov

Základné stavebné prvky každého programu sú údaje a algoritmus.

Údaje - abstrakcia reálneho sveta, vyjadrujú určité charakteristiky a vlastnosti reálnych objektov. Údaje, s ktorými program pracuje, majú určitú štruktúru.

Štruktúry údajov - sa obmedzovali na bit, slovo, pole slov. Stačili na reprezentáciu - čísla a logickej hodnoty.

Použitie ďalších štruktúr vyžaduje:

- ako reprezentovať objekty štruktúry v pamäti počítača
- vyjadriť operácie nad štruktúrami, pomocou operácií, ktoré poskytuje počítač

Aplikácie údajových typov a štruktúr pri tvorbe algoritmov

Sčítaním dvoch vektorov dostaneme opäť vektor - pripúšťame iba také operácie, ktoré nemenia typ organizácie - typ štruktúry - údajový typ (typ údajov).

Typ údajov - charakterizuje určitú množinu, ktorej prvky sa nazývajú hodnotami daného typu - typ údajov určuje množinu hodnôt, do ktorej daná konštanta prináleží, alebo ktoré môže daná premenná či výraz nadobudnúť.

Typ hodnoty konštanty, premennej, výrazu sa dá určiť z ich deklarácie.

Klasifikácia údajových typov

1. Jednoduché údajové typy
   a) celé čísla
   b) reálne čísla
   c) logické hodnoty
   d) znaky

2. Zložené údajové typy:
   a) vektor
   b) pole
   c) záznam
   d) množina
   e) reťazec
   f) zásobník

1. Jednoduché údajové typy (neštruktúrované)

a) celé čísla (integer)

Majú reprezentáciu v 1 slove (n-bitov). Príslušný vektor bitov sa interpretuje ako zápis celého čísla v dvojkovej sústave v pevnej rádovej čiarke (priamy kód, inverzný kód, doplnkový kód). Operácie - základné aritmetické operácie.

b) reálne čísla (real)

Môžu byť implementované:

- zobrazením v pevnej rádovej čiarke - ohraničená absolútna chyba
- zobrazením v pohyblivej rádovej čiarke (ohraničená relatívna chyba)

Základné aritmetické operácie - je výhodné, ak sú v danom počítači definované odpovedajúce strojové operácie.

1. Jednoduché údajové typy

c) logická hodnota

Pre efektívnu implementáciu - najmenšia adresovateľná časť pamäte - n-bitový vektor (n=8).

d) znaky (char)

Pri reprezentácii údajov typu char sa využíva kód na zobrazenie von. abecedy používaný na danom PC. Údaje sa reprezentujú v 8-bitovom vektore (slabike) ako celé čísla.

Je potrebné zvoliť primeraný spôsob interpretácie vektora bitov v súlade s možnosťami PC.

2. Zložené údajové typy (štruktúrované)

Je možné ich rozdeliť na menšie plnovýznamové celky. Ide o objekty, ktoré sú zložené zo skalárnych typov ich združením.

a) vektor

Usporiadaná n-tica prvkov, z ktorej každý prvok je prístupný pomocou celého čísla - index. Typ údaja vektor - je motivovaný štruktúrou a organizáciou hlavnej pamäte PC.

Hlavná pamäť - považujeme za vektor pamäťových miest a adresu pamäťového miesta za index prvku vektora (aj pamäťové miesto môžeme považovať za vektor bitov).

Zreťazená voľná pamäť - dynamické typy údajov (počet prvkov údaja sa po vykonaní operácie mení).

2. Zložené údajové typy (štruktúrované)

b) pole

Pole je homogénna štruktúra, ktorá pozostáva z prvkov, ktoré sú všetky jedného typu (pole je štruktúra s náhodným prístupom).

Na referenciu individuálneho prvku poľa je potrebné použiť meno celej štruktúry rozšírené o index, určujúci vybratý prvok. Index nadobúda hodnoty typu definovaného ako typ indexu poľa (príklady na cvičeniach - algoritmy triedenia, max. z množiny čísel).

A(I) - jednorozmerné pole (prvkom poľa je číslo, znak)

A(I,J) - dvojrozmerné pole (prvkom poľa je iné pole)

2. Zložené údajové typy (štruktúrované)

c) záznam (record)

Záznam je štruktúrovaný údaj, ktorý pozostáva zo zložiek rôzneho typu (heterogénne pole). Je to statická štruktúra a implementuje sa pomocou vektora.

Implementácia záznamu zahrňuje realizáciu sprístupnenia jednotlivých zložiek, ktoré majú rôznu dĺžku.

d) množina

Množina je štruktúrovaný typ zhodný s interpretáciou v matematike, prvky sú konečné množiny a operácie sú množinové operácie (prienik, zjednotenie, rozdiel, ...).

2. Zložené údajové typy (štruktúrované)

e) reťazec

Je to usporiadaná n-tica prvkov danej množiny.

f) zásobník

Zásobník je štruktúrovaný typ údajov, ktorý možno zo statického hľadiska stotožniť s reťazcom (postupnosť prvkov, ktorá sa mení iba na jednom konci).

Spôsoby reprezentácie údajových typov v programoch

Typ - určuje množinu prípustných hodnôt a reprezentáciu údaju v programe (nešpecifikuje konkrétne hodnoty jednotlivých prvkov objektu).

Najjednoduchšie objekty, s ktorými pri tvorbe algoritmov (programu) pracujeme:

I. konštanty

II. premenné

Údajové objekty

I. konštanty

Údajový objekt, ktorý sa počas realizácie algoritmu (počas behu programu) nemení.

Konštanty sú charakterizované jednoznačne svojou hodnotou.

II. premenné

Údajový objekt, ktorý počas svojej existencie v algoritme (programe) mení svoju hodnotu.

Premenné ~ k pamäťovému miestu - vzhľadom na možnosť zmeny ich hodnoty musia mať špecifikované umiestnenie v pamäti - na označenie premenných použijeme identifikátor.

Násobenie vektora konštantou

Príklad 1 - Násobenie vektora konštantou

Navrhnite algoritmus násobenia vektora konštantou.

Jednorozmerné pole - indexovaná premenná A(I)

matematicky: $b_i = K \cdot a_i$, $i = 1,2,\ldots,n$

algoritmicky: B(I)←K*A(I), I=1,2,…,N

Analýza úlohy:

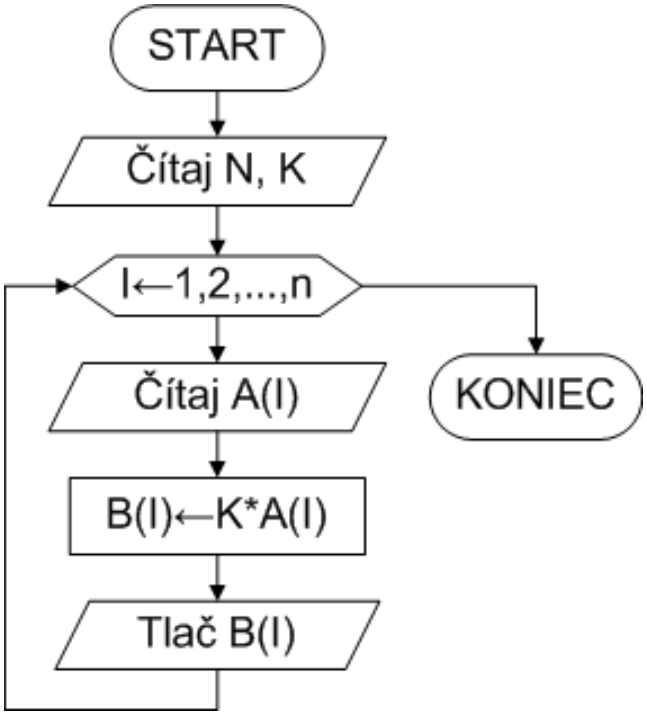
Použijeme cyklus s daným počtom opakovaní pre načítanie prvkov poľa a aj pre vykonanie násobenia vektora konštantou pre ∀i

Vývojový diagram na nasledujúcej strane >>>

Násobenie vektora konštantou

Príklad 1 - Algoritmus pre násobenie vektora konštantou

Vstupy: I - syst. premenná
K - konštanta
N - počet prvkov vektora
Výstupy: vektor B(I) I=1, N



Súčet dvoch vektorov

Príklad 2 - Súčet dvoch vektorov

Navrhnite algoritmus pre súčet dvoch vektorov.

Pre súčet dvoch vektorov (rovnaký rozmer) v matematike platí:

$$Ci=ai+bi,\ i=1,2,\ldots,n$$

postupne sa sčítajú odpovedajúce hodnoty prvkov obidvoch vektorov.

Vstupy: I - riadiaca premenná cyklu
M, N - rozmery vektorov
A(I) - vektor A, B(I) - vektor B

Výstup: C(I) - vektor C a jeho zložky

Vývojový diagram na nasledujúcej strane >>>

Súčet dvoch vektorov

Príklad 2 - Algoritmus pre súčet dvoch vektorov