

Informatika – Algoritmy

Radim Farana

Podklady předmětu Informatika
pro akademický rok 2010/2011

Obsah

- Algoritmus.
- Vlastnosti algoritmu.
- Popis algoritmu.
- Hodnocení algoritmů.
- Příklady algoritmů.

Algoritmus

- **Algoritmus** je přesný předpis definující výpočtový proces vedoucí od měnitelných výchozích údajů až k žadaným (vždy správným) výsledkům. Tento předpis se skládá z jednotlivých výpočtových kroků, které jsou zapsány v určitém pořadí. Počet výpočtových kroků musí být konečný.



Muhammad ibn
Musa
Al'Khwarizmi
* 778? Baghdad
+ 850?
<http://www.cs.pu.edu.au/~rwh/teaching/Subject/cp1030/1997/lectures/historyofalgor.htm>

Vlastnosti algoritmu

- **determinovanost** - shrnuje **přesnost**, **srozumitelnost** a **jednoznačnost**. V každém okamžiku řešení musí být jasné, jakou operaci má algoritmus provádět.
- **hromadnost** (masovost) - algoritmus musí popisovat zpracování celé skupiny příbuzných hodnot.
- **rezultativnost** - algoritmus musí vždy dospět ke správnému výsledku, a to pomocí **konečného** počtu kroků.
- **opakovatelnost** - při stejných hodnotách vstupních dat musí algoritmus vždy dospět ke stejnému výsledku.

Algoritmus versus program

- **program** = posloupnost příkazů,
 - dokumentu je se výpisem programu,
 - je chráněn autorským zákonem.
- **algoritmus** = postup práce,
 - dokumentuje se zápisem algoritmu,
 - je možné ho patentovat.
- Program realizuje algoritmus (algoritmy), algoritmus je jeho nutnou součástí.

Popis algoritmu

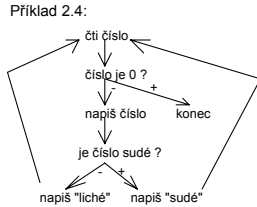
- **Slovní popis**
 - pracovní postup,
 - strukturovaný text, zápis pomocí grafu,
 - pseudokód (programovací).
- **Grafický zápis**
 - vývojový diagram,
 - Kopenogram, NS-diagram,
 - strukturogram.

Slovní popis

Příklad 2.1:
Z klávesnice čteme celá čísla a vypisujeme je na obrazovku doplněná o informaci, zda je číslo sudé nebo liché. Práce končí po vstupu čísla 0, které se nezpracovává.

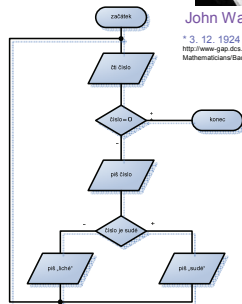
Příklad 2.2:
1 přečti číslo ze vstupu
2 když je číslo 0 jdi k bodu 9
3 vyšli číslo na výstup
4 když je číslo sudé, jdi k bodu 7
5 vyšli na výstup "liché"
6 jdi k bodu 1
7 vyšli na výstup "sudé"
8 jdi k bodu 1
9 konec

Příklad 2.3:
10 Read Císto
20 If Císto=0 Then Goto 90
30 Write Císto
40 If Int(Císto/2)*2=Císto Then Goto 70
50 Write "liché"
60 Goto 10
70 Write "sudé"
80 Goto 10
90 End



Vývojový diagram

- Popis algoritmů pro FORTRAN (FORMula TRANslator)
- IBM v r. 1954
- Formalizován různými normativy (ČSN 36 9030)



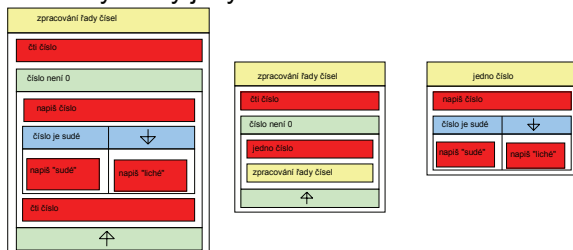
John Warner Baskus
* 3. 12. 1924 Philadelphia, USA
<http://www.gap.dcs.fr.ac.uk/~history/Mathematicians/Baskus.html>

Kopenogram

Ing. Rudolf Pecinovský, CSc.
* 17. 7. 1954, Praha
<http://rudoif.pecinovsky.cz/>
MUDr. Jiří Kofránek, CSc.



- Autoři: Kofránek, Pecinovský a Novák
- Pro výukový jazyk Karel



N-S diagramy

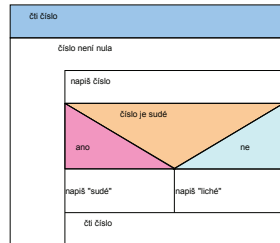
- Autoři:
Nassi a Schneiderman
- Úspornější zápis
- Stále používán
- Plug-in do Vizio



Isaac "Ike" Nassi
<http://www.nassi.com>

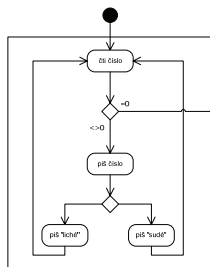


Ben Schneiderman
* 21. 8. 1947
<http://www.cs.umsl.edu/~ben/>

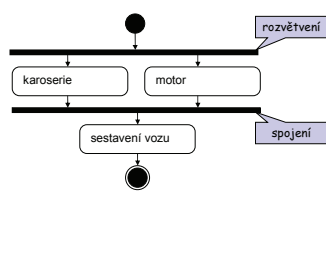


Diagramy aktivit UML

Zpracování souboru dat



Podpora paralelismu



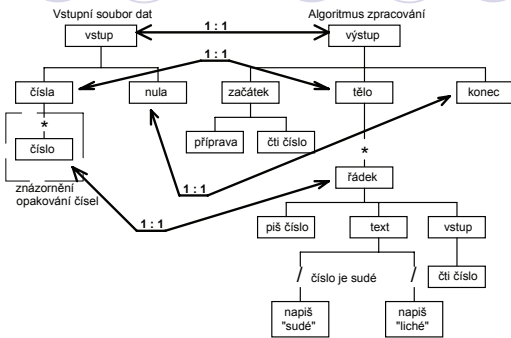
Strukturogramy

- M. A. Jackson, 1975
- Základní struktury:
 - sekvence (posloupnost operací),
 - selekce (větvení).
 - opakování – zvláštní případ sekvence.
- Snadné postupné upřesňování algoritmu
- Jednoznačný vztah mezi daty a algoritmem



Michael Anthony Jackson
* 1936
<http://mca.open.ac.uk/myj66/>

Strukturogramy



Základní struktury

Pseudojazyk	Vývojový diagram	Kopenogram	Strukturogram
Programový celek (rutina, podprogram, procedura a pod.) - definice			
název (parametry) . . end název		 horní pruh vybarven žlutě	jeden celý strukturogram
programový celek - použití (volání)			
název (parametry)		 blok vybarven červeně, při rekurzivním volání žlutě	

Podmíněná činnost, rozhodování

Pseudojazyk	Vývojový diagram	Kopenogram	Strukturogram
Provedení konkrétní činnosti			
popis činnosti		 blok vybarven červeně	
Podmíněná činnost (provádí se pouze pokud je splněna určitá podmínka)			
if podmínka podmíněná činnost end if		 horní pruh vybarven modře	
Rozhodování (pokud platí určená podmínka, provede se činnost 1, jinak činnost 2)			
if podmínka činnost 1 else činnost 2 end if		 horní pruh vybarven modře	

Hodnocení algoritmů

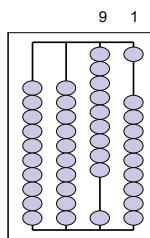
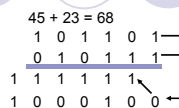
- Složitost algoritmu (binární bitová).
- Sčítání dvou k -bitových čísel má binární bitovou složitost úměrnou délce k .
- Násobení dvou k a j bitových čísel představuje Tedy celkem nejvýše $(j - 1)$ součtů $(k + j - 1)$ místných čísel:
 $(j - 1)(k + j - 1) < j(k + j) < 2kj$.
- Zkráceně: $f(n) = O(g(n))$,
 pro konečnou limitu: $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)}$

Složitost algoritmů

- **Polynomická složitost:** výpočet vyžaduje $O(k^d)$ bitových operací
 - sčítání $d = 1$,
 - násobení $d = 2$.
- **Nepolynomická složitost,** složitost výpočtu s rostoucím n roste rychleji.
 - $n!$ - $O(n \log_2^2(n)) = O(2^k k^2)$.
 - a^n

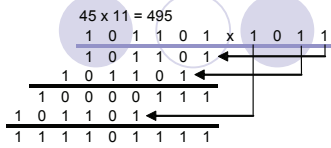
Realizace základních operací

- Sčítání (odečítání)
binárně: $45 + 23$
- dekadicky: $68 + 23$

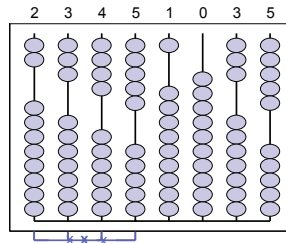


Násobení

- Násobení postupné sčítání a rotace



- dekadicky: 23.45



Realizace funkcí

- Taylorova řada

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a)$$

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots$$

$$\ln x = 2 \cdot \left[\frac{(x-1)}{(x+1)} + \frac{(x-1)^3}{3(x+1)^3} + \frac{(x-1)^5}{5(x+1)^5} + \dots \right]$$

