

Algoritmy

BI-PA1 – Programování a Algoritmizace I.

Ladislav Vagner

Katedra teoretické informatiky
Fakulta informačních technologií
ČVUT v Praze
`xvagner@fit.cvut.cz`

25. září 2013

Kontakt

- místnost A-1233, KTI FIT,
- e-mail: `xvagner@fit.cvut.cz`,
- agenda: prosemináře, Progtest.

Programovací jazyky a styly

- Imperativní (procedurální) programování:
 - Pascal, strojový kód, ...
- Funkcionální programování:
 - Haskell, Lisp, ...
- Logické (deklarativní) programování:
 - Prolog, SQL, XSLT, ...
- OO programování:
 - Eiffel, Objective C, Smalltalk, ...

Programovací jazyky a styly

- Imperativní (procedurální) programování:
 - Pascal, strojový kód, ...
- Funkcionální programování:
 - Haskell, Lisp, ...
- Logické (deklarativní) programování:
 - Prolog, SQL, XSLT, ...
- OO programování:
 - Eiffel, Objective C, Smalltalk, ...
- Kam patří Java, C++, a C?

Vlastnosti algoritmu:

- Hromadnost (univerzálnost).
- Jednoznačnost (determinismus).
- Poskytuje výsledky (resultativnost).
- Konečnost.
- Vstupy.
- Výstupy.
- Složitost.

Zápis algoritmu:

- textová podoba (pseudokód, programovací jazyk),
- grafická podoba (vývojový diagram – flow chart).

Algoritmy

Základní elementy při zápisu algoritmu:

- vstupní bod,
- koncový bod,
- větvení,
- příkaz.

Odvozené:

- smyčky (= větvení + návrat),
- I/O operace (= speciální typ příkazu),
- vyvolání jiného algoritmu (= speciální typ příkazu).

Algoritmy – vývojové diagramy

Elementy vývojového diagramu:

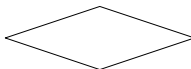
Vstupní bod



Koncový bod



Podmínka (větvení)



Příkaz

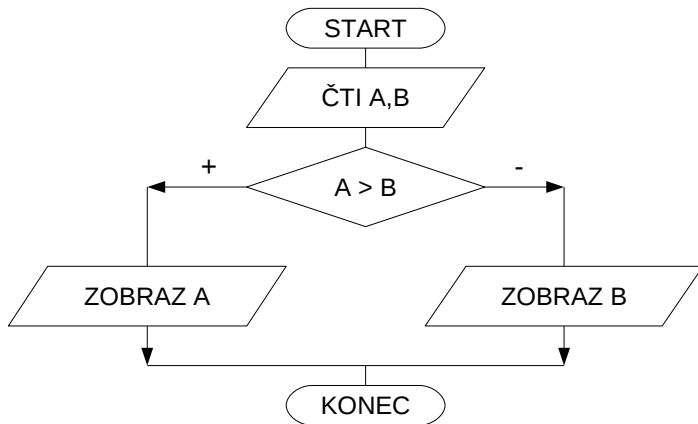


I/O operace



Algoritmy – vývojové diagramy

Příklad: algoritmus načte dvě čísla a zobrazí větší z nich.



Algoritmy – pseudokód

Příklad: algoritmus načte dvě čísla a zobrazí větší z nich.

START

ČTI A, B

POKUD $A > B$

 ZOBRAZ A

JINAK

 ZOBRAZ B

KONEC

Algoritmy – kvadratická rovnice

Úkol: vymyslet a zapsat algoritmus, který dokáže vyřešit zadanou kvadratickou rovnici.

Algoritmy – kvadratická rovnice

START

ČTI a, b, c

$d := b * b - 4 * a * c$

$x1 := (-b + \text{sqrt}(d)) / 2 / a$

$x2 := (-b - \text{sqrt}(d)) / 2 / a$

ZOBRAZ x1, x2

KONEC

Je tento algoritmus správný?

Algoritmy – kvadratická rovnice

START

ČTI a, b, c

$d := b * b - 4 * a * c$

$x1 := (-b + \text{sqrt}(d)) / 2 / a$

$x2 := (-b - \text{sqrt}(d)) / 2 / a$

ZOBRAZ $x1, x2$

KONEC

Je tento algoritmus správný?

Co se stane pro $a = 0$?

Algoritmy – kvadratická rovnice

START

ČTI a, b, c

$d := b * b - 4 * a * c$

$x1 := (-b + \text{sqrt}(d)) / 2 / a$

$x2 := (-b - \text{sqrt}(d)) / 2 / a$

ZOBRAZ $x1, x2$

KONEC

Je tento algoritmus správný?

Co se stane pro $a = 0$?

Co se stane pro $d < 0$?

Algoritmy – kvadratická rovnice

START

ČTI a, b, c

POKUD a = 0

 ZOBRAZ "Neni kvadraticka rovnice"

 KONEC

d := b * b - 4 * a * c

POKUD d < 0

 ZOBRAZ "Neexistuje realne reseni"

 KONEC

x1 := (-b + sqrt (d)) / 2 / a

x2 := (-b - sqrt (d)) / 2 / a

ZOBRAZ x1, x2

KONEC

Algoritmy – minimum, maximum a prostřední číslo

Úkol: pro tři zadaná čísla a , b a c (navzájem různá) určit maximum, minimum a prostřední číslo.

Algoritmy – minimum, maximum a prostřední číslo

```
START
ČTI a, b, c
POKUD a > b
    POKUD a > c
        max := a
    POKUD b > c
        mid := b
        min := c
    JINAK
        mid := c
        min := b
JINAK
    max := c
    mid := a
    min := b
JINAK
```

```
POKUD b > c
    max := b
    POKUD a > c
        mid := a
        min := c
    JINAK
        mid := c
        min := a
JINAK
    max := c
    mid := b
    min := a
ZOBRAZ "Maximum", max
ZOBRAZ "Prostredni", mid
ZOBRAZ "Minimum", min
KONEC
```


Algoritmy – minimum, maximum a prostřední číslo

```
START
ČTI a, b, c
max := a
POKUD b > max
    max := b
POKUD c > max
    max := c
min := a
POKUD b < min
    min := b
POKUD c < min
    min := c
mid := a + b + c - min - max
ZOBRAZ "Maximum", max
ZOBRAZ "Prostredni", mid
ZOBRAZ "Minimum", min
KONEC
```

Algoritmy – společný dělitel a násobek

Úkol: zapsat algoritmus, který pro zadaná přirozená čísla a a b určí jejich největší společný dělitel a nejmenší společný násobek.

Algoritmy – společný dělitel a násobek

```
START
ČTI a, b
prod := a * b
DOKUD a <> b
    POKUD a > b
        a := a - b
    JINAK
        b := b - a
gcd := a
lcm := prod / gcd
ZOBRAZ "Nejvetsi spolecny delitel", gcd
ZOBRAZ "Nejmensi spolecny nasobek", lcm
KONEC
```

Je tento algoritmus efektivní?

Algoritmy – společný dělitel a násobek

```
START
ČTI a, b
prod := a * b
DOKUD a <> b
    POKUD a > b
        a := a - b
    JINAK
        b := b - a
gcd := a
lcm := prod / gcd
ZOBRAZ "Nejvetsi spolecny delitel", gcd
ZOBRAZ "Nejmensi spolecny nasobek", lcm
KONEC
```

Je tento algoritmus efektivní?

Pro 60 a 36.

Algoritmy – společný dělitel a násobek

```
START
ČTI a, b
prod := a * b
DOKUD a <> b
    POKUD a > b
        a := a - b
    JINAK
        b := b - a
gcd := a
lcm := prod / gcd
ZOBRAZ "Nejvetsi spolecny delitel", gcd
ZOBRAZ "Nejmensi spolecny nasobek", lcm
KONEC
```

Je tento algoritmus efektivní?

Pro 60 a 36.

Pro 1001 a 1.

Algoritmy – společný dělitel a násobek

```
START
ČTI a, b
prod := a * b
POKUD a < b
    tmp := a
    a := b
    b := tmp
DOKUD b > 0
    tmp := a mod b
    a := b
    b := tmp
gcd := a
lcm := prod / gcd
ZOBRAZ "Nejvetsi spolecny delitel", gcd
ZOBRAZ "Nejmensi spolecny nasobek", lcm
KONEC
```

Algoritmy – maximum v posloupnosti

Úkol: zapsat algoritmus, který pro zadanou posloupnost n navzájem různých celých čísel najde maximum.

Algoritmy – maximum v posloupnosti

```
START
ČTI n
max := 0
DOKUD n > 0
    ČTI x
    POKUD x > max
        max := x
    n := n - 1
ZOBRAZ "Maximum", max
KONEC
```

Je algoritmus správný?

Algoritmy – maximum v posloupnosti

```
START
ČTI n
max := 0
DOKUD n > 0
    ČTI x
    POKUD x > max
        max := x
    n := n - 1
ZOBRAZ "Maximum", max
KONEC
```

Je algoritmus správný?

Ne pro $n = 3$ a vstup: -1, -2, -3.

Algoritmy – maximum v posloupnosti

```
START
ČTI n
POKUD n <= 0
    ZOBRAZ "Chyba ..."
    KONEC
ČTI max
n := n - 1
DOKUD n > 0
    ČTI x
    POKUD x > max
        max := x
    n := n - 1
ZOBRAZ "Maximum", max
KONEC
```

Algoritmy – druhé největší číslo

Úkol: zapsat algoritmus, který pro zadanou posloupnost n navzájem různých celých čísel najde druhé největší číslo.

Algoritmy – druhé největší číslo

```
START
ČTI n
POKUD n < 2
    ZOBRAZ "Chyba ..."
KONEC
ČTI m1, m2
POKUD m1 < m2
    tmp := m1
    m1 := m2
    m2 := tmp
n := n - 2
```

```
DOKUD n > 0
    ČTI x
    POKUD x > m1
        m2 := m1
        m1 := x
    n := n - 1
ZOBRAZ "Druhé maximum", m2
KONEC
```

Je algoritmus správný?

Algoritmy – druhé největší číslo

```
START
ČTI n
POKUD n < 2
    ZOBRAZ "Chyba ..."
KONEC
ČTI m1, m2
POKUD m1 < m2
    tmp := m1
    m1 := m2
    m2 := tmp
n := n - 2
```

```
DOKUD n > 0
    ČTI x
    POKUD x > m1
        m2 := m1
        m1 := x
    n := n - 1
ZOBRAZ "Druhé maximum", m2
KONEC
```

Je algoritmus správný?

Ne pro $n = 3$ a vstup: 1, 3, 2.

Algoritmy – druhé největší číslo

```
START
ČTI n
POKUD n < 2
    ZOBRAZ "Chyba ..."
KONEC
ČTI m1, m2
POKUD m1 < m2
    tmp := m1
    m1 := m2
    m2 := tmp
n := n - 2
```

```
DOKUD n > 0
    ČTI x
    POKUD x > m1
        m2 := m1
        m1 := x
    JINAK
        POKUD x > m2
            m2 := x
        n := n - 1
    ZOBRAZ "Druhé maximum", m2
KONEC
```

Je algoritmus správný?

Algoritmy – druhé největší číslo

```
START
ČTI n
POKUD n < 2
    ZOBRAZ "Chyba ..."
KONEC
ČTI m1, m2
POKUD m1 < m2
    tmp := m1
    m1 := m2
    m2 := tmp
n := n - 2
```

```
DOKUD n > 0
    ČTI x
    POKUD x > m1
        m2 := m1
        m1 := x
    JINAK
        POKUD x > m2
            m2 := x
        n := n - 1
    ZOBRAZ "Druhé maximum", m2
KONEC
```

Je algoritmus správný?

Ano.

Algoritmy – faktorizace

Úkol: zapsat algoritmus, který pro zadané přirozené číslo n určí jeho prvočíselný rozklad (faktORIZACI).

Např. pro $n = 720$ je faktORIZACE:

$$720 = 5 * 2 * 2 * 2 * 2 * 3 * 3$$

Algoritmy – faktORIZACE

```
START  
ČTI n  
i := 2  
DOKUD i < n  
    POKUD n mod i = 0  
        ZOBRAZ i  
        i := i + 1  
KONEC
```

Je algoritmus správně?

Algoritmy – faktORIZACE

```
START  
ČTI n  
i := 2  
DOKUD i < n  
    POKUD n mod i = 0  
        ZOBRAZ i  
        i := i + 1  
KONEC
```

Je algoritmus správně?

Ne, zobrazí všechny dělitele, ne pouze prvočísla.

Algoritmy – faktORIZACE

```
START
ČTI n
i := 2
DOKUD i < n
    POKUD n mod i = 0
        POKUD isprime ( i )
            ZOBRAZ i
        i := i + 1
KONEC
```

Je algoritmus správně?

Algoritmy – faktORIZACE

```
START
ČTI n
i := 2
DOKUD i < n
    POKUD n mod i = 0
        POKUD isprime ( i )
            ZOBRAZ i
        i := i + 1
KONEC
```

Je algoritmus správně?

Ne, zobrazí prvočíselné faktory, ale pouze jednou.

Algoritmy – faktORIZACE

```
START
ČTI n
i := 2
DOKUD i <= n
    POKUD n mod i = 0
        POKUD isprime ( i )
            ZOBRAZ i
            n := n / i
        JINAK
            i := i + 1
KONEC
```

Je algoritmus správně?

Algoritmy – faktorizace

```
START
ČTI n
i := 2
DOKUD i <= n
    POKUD n mod i = 0
        POKUD isprime ( i )
            ZOBRAZ i
            n := n / i
        JINAK
            i := i + 1
KONEC
```

Je algoritmus správně?

Ano, pouze pro $n = 1$ nezobrazí nic.

Algoritmy – faktorizace

```
START
ČTI n
i := 2
DOKUD i <= n
    POKUD n mod i = 0
        POKUD isprime ( i )
            ZOBRAZ i
            n := n / i
        JINAK
            i := i + 1
KONEC
```

Je algoritmus správně?

Ano, pouze pro $n = 1$ nezobrazí nic.

Je velmi neefektivní.

Algoritmy – faktorizace

```
START
ČTI n
i := 2
DOKUD i <= n
    POKUD n mod i = 0
        ZOBRAZ i
        n := n / i
    JINAK
        i := i + 1
KONEC
```

Je algoritmus správně?

Algoritmy – faktorizace

```
START
ČTI n
i := 2
DOKUD i <= n
    POKUD n mod i = 0
        ZOBRAZ i
        n := n / i
    JINAK
        i := i + 1
KONEC
```

Je algoritmus správně?

Ano, pouze pro $n = 1$ nezobrazí nic.

Algoritmy – faktORIZACE

```
START
ČTI n
i := 2
DOKUD i <= n
    POKUD n mod i = 0
        ZOBRAZ i
        n := n / i
    JINAK
        i := i + 1
KONEC
```

Je algoritmus správně?

Ano, pouze pro $n = 1$ nezobrazí nic.

Je stále neefektivní.

Algoritmy – faktORIZACE

START

ČTI n

factorize (n)

KONEC

factorize (n):

START

$i := \text{sqrt} (n)$

DOKUD $i \geq 2$

 POKUD $n \bmod i = 0$

 factorize (i)

 factorize (n / i)

 KONEC

$i := i - 1$

ZOBRAZ n

KONEC

Dotazy . . .

Děkuji za pozornost.