

5 Složitost a správnost algoritmů

5.1 Řešení příkladu z domácí přípravy.

Je dán následující algoritmus

```

i := N
while (i > 0) do {
  j := 0;
  i := ⌊i/3⌋;
  while (j < i) do write(★); j ++ }

```

Spočítejte časové nároky algoritmu, tj najděte nejjednodušší funkci $f(N)$ takovou, že časové nároky jsou v třídě $\Theta(f(N))$. (Samotná funkce nestačí, musíte zdůvodnit, proč je správně.)

Nové příklady

5.2 Jsou dána dvě kladná přirozená čísla m, n

```

x := m; y := n; z := 0
while x ≠ 0 do {
  if x (mod 2) = 1 then
    z := y + z
  x := ⌊x/2⌋;
  y := y · 2 }
return z

```

- Spočítejte časové nároky algoritmu, tj najděte funkci f takovou, že časové nároky jsou v třídě $\Theta(f)$.
- Zjistěte jakou hodnotu má proměnná z na konci práce algoritmu.
- Dokažte své tvrzení z bodu (b), tj. najděte invariant a dokažte ho.

5.3 Je dán následující algoritmus:

Vstup: Posloupnost s_1, s_2, \dots, s_n celých čísel mezi kterými je aspoň jedno číslo kladné.

Výstup: Číslo max .

```

procedure MAX(s, n)
  max := 0;
  sum := 0;
  for i = 1 to n do {
    begin
      if sum + si > 0 then
        sum := sum + si
      else
        sum := 0
      if sum > max then
        max := sum
    }
  return(max)

```

(a) Použijte algoritmus na vstup:

27 6 -50 21 -3 14 16 -8 42 33 -21 9.

(Tj. $n = 12$, $s_1 = 27$, $s_2 = 6$, atd.)

(b) Charakterizujte výstup algoritmu, tj. číslo *max*.

(c) Dokažte správnost vašich tvrzení z bodu b).

5.4 Rozhodněte, zda je správný algoritmus pro nalezení minimální kostry. V případě, že správný je, najděte variant a dokažte invariant, v případě, že správný není, najděte protipříklad.

```
minKostr(G = (V, E), a: E → ℤ)
T := ∅
foreach e ∈ E do
  if T ∪ {e} neobsahuje kružnici then
    T := T ∪ {e}
return T
```

5.5 Rozhodněte, zda je správný algoritmus pro nalezení minimální kostry. V případě, že správný je, najděte variant a dokažte invariant, v případě, že správný není, najděte protipříklad.

```
minKostr(G = (V, E), a: E → ℤ)
T := ∅
foreach e ∈ E do
  T := T ∪ {e}
  if T obsahuje kružnici C then
    T := T \ {e'}, kde e' je nejdražší hrana C
return T
```

5.6 Je dán následující algoritmus

Vstup: Přirozené kladné číslo x .

Výstup: $f(x)$.

```
y1 = x; y2 = 1
while y2 ≠ 0 do {
  if y1 > 100 then
    y1 = y1 - 10; y2 = y2 - 1
  else
    y1 = y1 + 11; y2 = y2 + 1
}
return y1
```

(a) Použijte algoritmus na vstup $x = 93$.

(b) Dokažte, že se algoritmus zastaví na každém vstupu. Odhadněte počet průchodů while cyklu pro jednotlivé vstupy.

(c) Charakterizujte výstup algoritmu, tj. hodnotu $f(x) = y_1$.

(d) Dokažte správnost vašeho tvrzení z bodu (c).

Výsledkem je číslo 91.