Algoritmizace: 1. cvičení

Matouš Vrba

20. 9. 2021

Organizace

- Matouš Vrba, matous.vrba@fel.cvut.cz, KN:E-119
- Zápočet za domácí úkoly zadané na přednáškách a účast na cvičení (max 2 absence)
- Konzultace možné po domluvě kdykoliv

Průběh cvičení

- Dvě části
  1. Shrnutí látky z poslední přednášky a společné řešení několika příkladů (cca 40 minut)
  2. Řešení vybraných příkladů v 1-4členných skupinách a odeslání cvičícímu (cca 50 minut)
- Příklady pro cvičení na webu předmětu (CourseWare: `https://cw.fel.cvut.cz/wiki/courses/b4b33alg/cviceni`)
  - K vyřešení je potřeba pouze tužka a papír (a možná někdy kalkulačka)
  - Nebudeme nic programovat (až u zkoušky)
- Na nejasnosti se ptejte hned – není to ostuda
- Na dotazy se snažte odpovídat – chyby jsou normální

Příklad 1

Který fragment programu z následujících dvou proběhne rychleji?
(Předpokládáme, že oba běží v identickém SW/HW prostředí.)

```
int n = 100;
int sum = 0;
for (i = 0; i < n; i++)
  for (j = 0; j < i; j++)
    sum += i+j;
```

```
int n = 75;
int sum = 0;
for (i = 0; i < n; i++)
  for (j = 0; j < n; j++)
    sum += i+j;
```

Příklad 1

Který fragment programu z následujících dvou proběhne rychleji?
(Předpokládáme, že oba běží v identickém SW/HW prostředí.)

```
int n = 100;
int sum = 0;
for (i = 0; i < n; i++)
  for (j = 0; j < i; j++)
    sum += i+j;
```

```
int n = 75;
int sum = 0;
for (i = 0; i < n; i++)
  for (j = 0; j < n; j++)
    sum += i+j;
```

Kolik iterací vnitřního cyklu proběhne?

Vlevo:

$$\begin{aligned} i = 0 &: & 0 \text{ iterací,} \\ i = 1 &: & 1 \text{ iterace,} \\ \dots & & \dots, \\ i = n-1 &: & n-1 \text{ iterací} \end{aligned}$$

Příklad 1

Který fragment programu z následujících dvou proběhne rychleji?
(Předpokládáme, že oba běží v identickém SW/HW prostředí.)

```
int n = 100;
int sum = 0;
for (i = 0; i < n; i++)
  for (j = 0; j < i; j++)
    sum += i+j;
```

```
int n = 75;
int sum = 0;
for (i = 0; i < n; i++)
  for (j = 0; j < n; j++)
    sum += i+j;
```

Kolik iterací vnitřního cyklu proběhne?

Vlevo: $\sum_{i=0}^{n-1} i = n\frac{n-1}{2} = 4950$.

Součet prvních q členů aritmetické posloupnosti: $\sum_{i=1}^{q} a_i = q\frac{a_1+a_q}{2}$.

Vpravo: $n^2 = 5625$.

Příklad 3

Do následujícího kódu doplňte chybějící konstantu v podmínce tak, aby byla procedura `xyz()` volána právě 2000 krát.

```
i = 50;
do {
  for (j=0; j < 70; j++) {
    if (j > ___) xyz();
  }
  i++;
} while (i < 150);
```

Příklad 3

Do následujícího kódu doplňte chybějící konstantu v podmínce tak, aby byla procedura `xyz()` volána právě 2000 krát.

```
i = 50;
do {
  for (j=0; j < 70; j++) {
    if (j > ___) xyz();
  }
  i++;
} while (i < 150);
```

Vnější cyklus vykoná 100 iterací. Protože $2000/100 = 20$, musí podmínka platit vždy ve 20 iteracích vnitřního cyklu. Z toho $70 - 20 = 50$ musí být první j, pro které bude podmínka platit, tj. chybějící konstanta je 49.

Příklad 6

Ke zpracování k-tého řádku matice velikosti $n \times n$ je zapotřebí $2k$ operací. Kolik je potřeba operací ke zpracování celé matice?

A. $2n^2$
B. $\frac{n^2}{2}$
C. $\frac{n(n+1)}{2}$
D. $n(n-1)$
E. $n(n+1)$

Příklad 6

Ke zpracování k-tého řádku matice velikosti $n \times n$ je zapotřebí $2k$ operací. Kolik je potřeba operací ke zpracování celé matice?

A. $2n^2$
B. $\frac{n^2}{2}$
C. $\frac{n(n+1)}{2}$
D. $n(n-1)$
E. $n(n+1)$

$$
\begin{aligned}
k = 1: & \quad 2 \text{ iterace,} \\
k = 2: & \quad 4 \text{ iterace,} \\
k = 3: & \quad 6 \text{ iterací,} \\
\cdots & \quad \cdots, \\
k = n: & \quad 2n \text{ iterací}
\end{aligned}
$$

Příklad 6

Ke zpracování k-tého řádku matice velikosti $n \times n$ je zapotřebí $2k$ operací. Kolik je potřeba operací ke zpracování celé matice?

A. $2n^2$
B. $\frac{n^2}{2}$
C. $\frac{n(n+1)}{2}$
D. $n(n-1)$
E. $n(n+1)$

Součet pro n řádků (aritmetická řada): $\sum_{k=1}^{n} 2k = n\frac{2+2n}{2} = n(n+1)$.

Součet prvních q členů aritmetické posloupnosti: $\sum_{i=1}^{q} a_i = q\frac{a_1+a_q}{2}$.

Příklad 7.A

Úloha, jejíž doba řešení je Cn^2, kde n je rozsah vstupních dat, se řeší na počítači pro $n = 5000$. Je zakoupen nový počítač, který je cca $2{,}5\times$ rychlejší. Jak lze zvětšit rozsah vstupních dat, aby byla úloha vyřešena na novém počítači ve stejném čase?

Příklad 7.A

Úloha, jejíž doba řešení je Cn^2, kde n je rozsah vstupních dat, se řeší na počítači pro $n = 5000$. Je zakoupen nový počítač, který je cca $2{,}5\times$ rychlejší. Jak lze zvětšit rozsah vstupních dat, aby byla úloha vyřešena na novém počítači ve stejném čase?

$$
\begin{aligned}
Cn_{new}^2 &= 2{,}5Cn_{old}^2 \\
n_{new} &= \sqrt{2{,}5 \cdot 5000^2}
\end{aligned}
$$

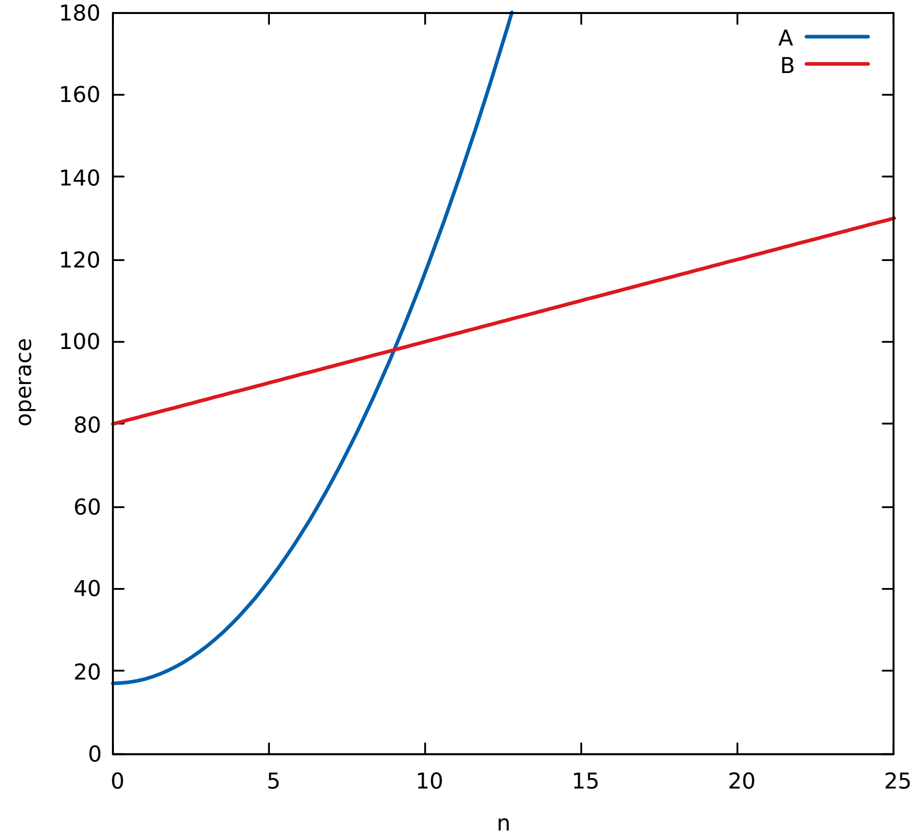
Tj. úloha lze nově vyřešit pro $n = 7905$ (po zaokrouhlení dolů).

Příklad 9

Metoda A potřebuje k vyřešení úlohy $n^2 + 17$ operací, Metoda B potřebuje $2n + 80$ operací, přičemž celé číslo n popisuje rozsah vstupních dat. Pro jaká n je výhodnější použít metodu A?

Příklad 9

Metoda A potřebuje k vyřešení úlohy $n^2 + 17$ operací, Metoda B potřebuje $2n + 80$ operací, přičemž celé číslo n popisuje rozsah vstupních dat. Pro jaká n je výhodnější použít metodu A?

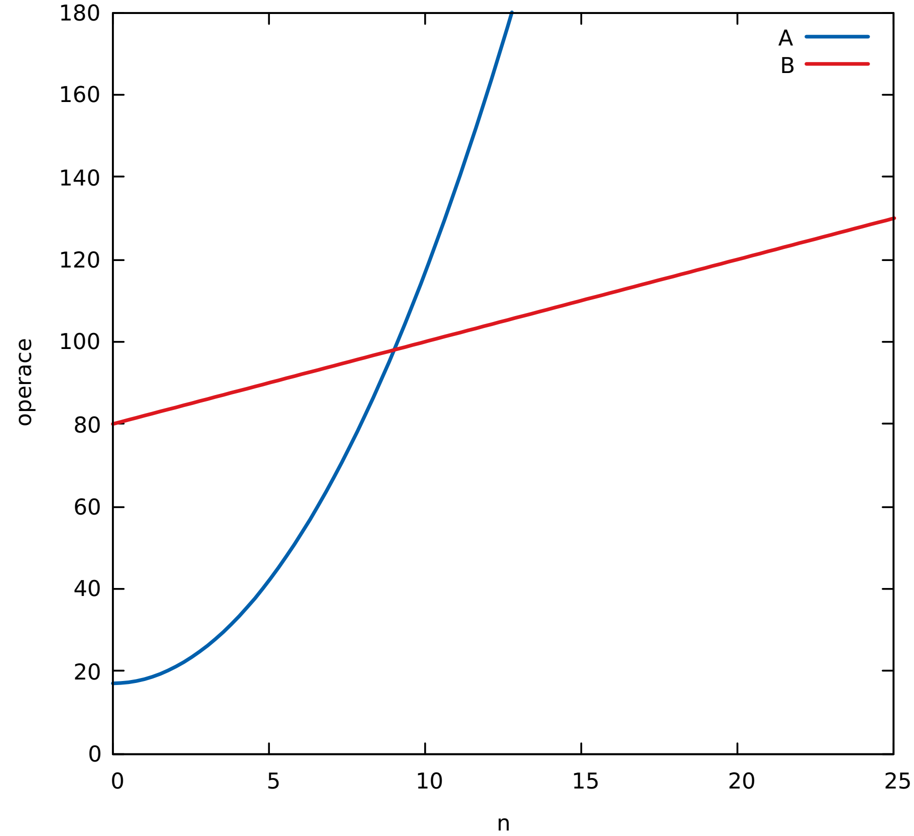


Příklad 9

Metoda A potřebuje k vyřešení úlohy $n^2 + 17$ operací, Metoda B potřebuje $2n + 80$ operací, přičemž celé číslo n popisuje rozsah vstupních dat. Pro jaká n je výhodnější použít metodu A?

Kde je průsečík?

$$\begin{aligned} n^2 + 17 &= 2n + 80 \\ n^2 - 2n - 63 &= 0 \\ (n+7)(n-9) &= 0 \\ n &= 9 \end{aligned}$$



Příklad 11

Každý ze dvou seznamů čísel je uspořádán v neklesajícím pořadí. Popište, jak vytvoříte třetí seznam, který bude obsahovat pouze taková čísla, která se vyskytují v obou daných seznamech. Musí vám stačit jeden průchod každým z daných seznamů.

Příklad 11

Každý ze dvou seznamů čísel je uspořádán v neklesajícím pořadí. Popište, jak vytvoříte třetí seznam, který bude obsahovat pouze taková čísla, která se vyskytují v obou daných seznamech. Musí vám stačit jeden průchod každým z daných seznamů.

seznam 1: $\big(1,\ \ 2,\ \ 3,\ \ 5,\ \ 7,\ \ 11,\ \ 13,\ \ 17,\ \ 19\big)$

seznam 2: $\big(1,\ \ 3,\ \ 5,\ \ 7,\ \ 9,\ \ 11,\ \ 13,\ \ 15,\ \ 17\big)$

seznam 3: $\big(\ \ ,\ \ \ \ ,\ \ \ \ ,\ \ \ \ ,\ \ \ \ ,\ \ \ \ ,\ \ \ \ ,\ \ \ \ ,\ \ \ \ \big)$

Příklad 13

Seznam obsahuje $N + 1$ celých čísel, každé leží v intervalu $[1, N]$, čísla nejsou v seznamu nijak uspořádána. Víme, že v seznamu se vyskytuje jedno číslo dvakrát, ostatní jen jednou. Určete duplikované číslo. Musí vám stačit jeden průchod seznamem a konstantně velká přidaná paměť.

Příklad 13

Seznam obsahuje $N+1$ celých čísel, každé leží v intervalu $[1, N]$, čísla nejsou v seznamu nijak uspořádána. Víme, že v seznamu se vyskytuje jedno číslo dvakrát, ostatní jen jednou. Určete duplikované číslo. Musí vám stačit jeden průchod seznamem a konstantně velká přidaná paměť.

Označme S součet čísel v seznamu a d duplikované číslo. Pak

$$S = \sum_{i=1}^{d} i + \sum_{i=d}^{N} i = d\frac{d+1}{2} + (N-d+1)\frac{N+d}{2},$$

a tudíž

$$d = \frac{2S - N(n+1)}{2}.$$

Samostatná práce

Následuje rozdělení do skupin po 1 až 4 členech.

Stáhněte si prezentaci s příklady: `https://bit.ly/2EtV7NY`.

Ve skupinách řešte příklady 5, 8A, 10, 12, 14 (v libovolném pořadí).

Výsledky mi pošlete na konci cvičení na email (matous.vrba@fel.cvut.cz) spolu se jmény členů skupiny.