Korrekthet och komplexitet

Simon Jutvreten
simonju@kth.se

16 november 2023

1 Multiplikation

I denna text antas att multiplikationsinstruktionen har tidskomplexiteten $O(1)$. Detta stämmer bra för multiplikation av små tal, men för större tal är komplexiteten till större del beroende av vilken algoritm som används för att beräkna produkten.

2 Uppgift 1: Induktion

2.1 Deluppgift 1

Bevisa $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$, för alla $n \in \mathbb{Z}_+$ med induktion.

Låt $VL = \sum_{i=1}^{n} i^2$ och $HL = \frac{n(n+1)(2n+1)}{6}$.

Induktionsbas:

$$\begin{aligned} n = 1 \implies VL_1 &= \sum_{i=1}^{1} i^2 \\ &= 1^2 \\ &= \frac{6}{6} \\ &= \frac{1(1+1)(2 \cdot 1 + 1)}{6} \\ &= HL_1 \\ \therefore VL_1 &= HL_1 \end{aligned}$$

Induktionsantagande (I):
Antag att $VL = HL$ för något $n = p$. Alltså, $\sum_{i=1}^{p} i^2 = \frac{p(p+1)(2p+1)}{6}$.

Induktionssteg:

$$
\begin{aligned}
n = p+1 \implies VL_{p+1} &= \sum_{i=1}^{p+1} i^2 \\
&= \sum_{i=1}^{p} i^2 + (p+1)^2 \\
&= \frac{p(p+1)(2p+1)}{6} + (p+1)^2 && \text{[I]} \\
&= \frac{(p+1)(p[2p+1]+6[p+1])}{6} \\
&= \frac{(p+1)(2p^2+p+6p+6)}{6} \\
&= \frac{(p+1)2(p^2+\frac{7}{2}p+3)}{6} \\
&= \frac{(p+1)2([p+\frac{7}{4}] - \frac{49}{16} + \frac{48}{16})}{6} \\
&= \frac{(p+1)2([p+\frac{7}{4}] - (\frac{1}{4})^2)}{6} \\
&= \frac{(p+1)2(p+2)(p+\frac{3}{2})}{6} \\
&= \frac{(p+1)([p+1]+2)(2[p+1]+1)}{6} \\
&= HL_{p+1} \\
\therefore VL_p = HL_p \implies VL_{p+1} &= HL_{p+1}
\end{aligned}
$$

Slutsats:
$VL_1 = HL_1$ och $VL_p = HL_p \implies VL_{p+1} = HL_{p+1}$ implicerar $VL_n = HL_n$, enligt induktionsprincipen. Således är $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$, för alla $n \in \mathbb{Z}_+$, v.s.b.

2.2 Deluppgift 2

Bevisa $\sum_{j=1}^{n}(2j-1) = n^2$, för alla $n \in \mathbb{Z}_+$ med induktion.

Låt $VL = \sum_{j=1}^{n}(2j-1)$ och $HL = n^2$.

Induktionsbas:

$$
\begin{aligned}
n = 1 \implies VL_1 &= \sum_{j=1}^{n} (2j - 1) \\
&= (2 \cdot 1 - 1) \\
&= 1 \\
&= 1^2 \\
&= HL_1 \\
\therefore VL_1 &= HL_1
\end{aligned}
$$

Induktionsantagande (II):
Antag att $VL = HL$ för något $n = p$. Alltså, $\sum_{j=1}^{p} (2j - 1) = p^2$.

Induktionssteg:

$$
\begin{aligned}
n = p + 1 \implies VL_{p+1} &= \sum_{j=1}^{p+1} (2j - 1) \\
&= \sum_{j=1}^{p} (2j - 1) + 2p + 1 \\
&= p^2 + 2p + 1 && \text{[II]} \\
&= (p + 1)^2 \\
&= HL_{p+1} \\
\therefore VL_p = HL_p \implies VL_{p+1} &= HL_{p+1}
\end{aligned}
$$

Slutsats:
$VL_1 = HL_1$ och $VL_p = HL_p \implies VL_{p+1} = HL_{p+1}$ implicerar $VL_n = HL_n$, enligt induktionsprincipen. Således är $\sum_{j=1}^{n} (2j - 1) = n^2$, där $n \in \mathbb{Z}_+$, v.s.b.

3 Uppgift 2: Iterativ korrekthet

Bevisa funktionens korrekthet och ange dess tidskomplexitet.

```
double expIterative(double x, int n) {
    double res = 1.0;

    for (int i = 0; i < n; i++) {
        res *= x;
    }

    return res;
}
```

Sätter in en loopinvariant i funktionen, och skriver om funktionen till en ekvivalent `while`-variant för att tydliggöra när `i` ökar i förhållande till loopinvarianten.

```
double expIterative(double x, int n) {
    double res = 1.0;

    {
        int i = 0;
        // Loop invariant: res = x^1 (initialization, A)

        while (i < n) {
            // Loop invariant: res = x^i (maintenance, B)

            res *= x;
            i++;
        }

        // Loop invariant: res = x^i (termination, C)
    }

    return res;
}
```

Bevisar att loopinvarianten är sann före- (A), i- (B) och efter loopen (C):

A (initialization):

$$i = 0 \implies res_1 = 1.0 = x^0 \text{ ok!}$$

B (maintenance):
Antag att loopinvarianten är sann för något $i = p$ (III). Alltså,

$$i = p \implies res_p = x^p$$

För att loopinvarianten skall vara sann iterationen efter, måste också

$$i = p + 1 \implies res_{p+1} = x^{p+1}$$

Söker påståendets sanningsvärde:

$$\begin{aligned} i = p + 1 \implies res_{p+1} &= res_p * x \\ &= x^p * x && \text{[III]} \\ &= x^{p+1} \\ \therefore res_p = x^p \implies res_{p+1} &= x^{p+1} \end{aligned}$$

Alltså, eftersom $res_1 = 1.0$ kommer $res_2 = x$, $res_3 = x^2$, och så vidare. Således är loopinvarianten sann för alla iterationer upp till och med $i = n - 1$.

C (termination):
Inna loopen avslutats sätts `res` till x^n (följer från B) och `i` till `n`, vilket uppfyller loopinvarianten vid C. Värdet som returneras är således x^n, vilket skulle bevisas.

Algoritmens tidskomplexitet:
Eftersom algoritmen utför multiplikationssinstruktionen `n` gånger, och instruktionen har tidskomplexiteten $O(1)$, är tidskomplexiteten av algoritmen $O(n)$.

4 Uppgift 3: Rekursiv korrekthet

Bevisa funktionens korrekthet och ange dess tidskomplexitet.

```
double expRecursive(double x, int n) {
    if (n <= 4) {
        return expIterative(x, n);
    }

    return expRecursive(x, n/2) * expRecursive(x, (n + 1)/2);
}
```

Notera att eftersom `n` är av typen `int` kan $\frac{n}{b}$ skrivas som $\lfloor \frac{n}{b} \rfloor$, där b är ett positivt heltal. Till följd av att `expIterative()` är bevisad kommer algoritmen vara korrekt för alla $0 \leq n \leq 4$. Det som sökes är sanningsvärdet för påståendet $F(x, n) = x^n \,\forall n \in \mathbb{Z}_+$, där $F(x, n) = \mathit{expRecursive}(x, n)$. Känt är

$$F(x,n) = \begin{cases} x^n & \text{om } 0 \leq n \leq 4 \\ F(x, \lfloor \frac{n}{2} \rfloor) \cdot F(x, \lfloor \frac{n+1}{2} \rfloor) & \text{om } n \geq 5 \end{cases}$$

Utför induktionsbevis:

Induktionsbas:

$$\begin{aligned} n = 5 \implies F(x,5) &= F(x, \left\lfloor \frac{5}{2} \right\rfloor) \cdot F(x, \left\lfloor \frac{6}{2} \right\rfloor) \\ &= F(x,2) \cdot F(x,3) \\ &= x^2 \cdot x^3 \\ &= x^5 \text{ ok!} \end{aligned}$$

Induktionsantagande (IV):
Antag att $F(x, 0) = 1$, $F(x, 1) = x$, $\ldots$, $F(x, p) = x^p$, för något $n = p$.

Induktionssteg:
Låt $n = p + 1$.
Fall $2 \mid p + 1$:

$$\begin{aligned} F(x, p+1) &= F(x, \left\lfloor \frac{p+1}{2} \right\rfloor) \cdot F(x, \left\lfloor \frac{p+2}{2} \right\rfloor) \\ &= [F(x, \left\lfloor \frac{p+1}{2} \right\rfloor)]^2 \\ &= (x^{\frac{p+1}{2}})^2 && \text{[IV]} \\ &= x^{p+1} && \text{ok!} \end{aligned}$$

Fall $2 \nmid p+1$:

$$\begin{aligned}
F(x, p+1) &= F(x, \left\lfloor \frac{p+1}{2} \right\rfloor) \cdot F(x, \left\lfloor \frac{p+2}{2} \right\rfloor) \\
&= F(x, \left\lfloor \frac{p}{2} \right\rfloor) \cdot F(x, \left\lfloor \frac{p}{2} + 1 \right\rfloor) \\
&= x^{\frac{p}{2}} \cdot x^{\frac{p}{2}+1} && \text{[IV]} \\
&= x^{p+1} && \text{ok!}
\end{aligned}$$

$$\therefore F(x,0) = 1 \land F(x,1) = x \land \cdots \land F(x,p) = x^p \implies F(x,p+1) = x^{p+1}$$

Slutsats:
Enligt induktionsprincipen är algoritmen korrekt för alla $n \in \mathbb{Z}_+$ och $x \in \mathbb{R}$.

Algoritmens tidskomplexitet:
Algoritmen delar upp problemet i $a = 2$ separata delar, nämligen `expRecursive(x, n/2)` och `expRecursive(x, (n+1)/2)`. Varje delproblem är ungefär $\frac{n}{2}$ stort vilket ger $b = 2$. Multiplikationen har tidskomplexiteten $f(n) = \Theta(1)$. Mästarsatsen, där $f(n) = \Theta(n^d)$, ger

$$\begin{aligned}
T(n) &= aT(\frac{n}{b}) + f(n) \\
&= \begin{cases} \Theta(n^d) & \text{om } a < b^d \\ \Theta(n^d \log n) & \text{om } a = b^d \\ \Theta(n^{\log_b a}) & \text{om } a > b^d \end{cases} \\
&= \Theta(n^{\log_b a}), \text{ ty } f(n) = \Theta(1) = \Theta(n^0) \implies d = 0 \\
&= \Theta(n^{\log_2 2}) \\
&= \Theta(n)
\end{aligned}$$

Alltså, den genomsnittliga tidskomplexiteten är $\Theta(n)$.

5 Uppgift 4: Tidigare läxa

Jag valde att arbeta vidare på min kompilator.