IndaPlusPlus - HW6

Uppgift 1.1

Bevisa med induktion att:

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

1. Basfall

Jag visar att det gäller för $n = 1$.

$$\mathrm{VL} = 1^1 = 1$$

$$\mathrm{HL} = \frac{1(1+1)(2 \cdot 1 + 1)}{6} = \frac{6}{6} = 1$$

$$\mathrm{VL} = \mathrm{HL}$$

Det stämmer för basfallet $n = 1$.

2. Induktionsantagande

Antag att det stämmer för $n = k$

$$\sum_{i=1}^{k} i^2 = \frac{k(k+1)(2k+1)}{6}$$

3. Induktionssteg

Visa att det stämmer för $n = k + 1$

$$\sum_{i=1}^{k+1} i^2 = \frac{(k+1)((k+1)+1)(2(k+1)+1)}{6}$$

$$\mathrm{VL} = \sum_{i=1}^{k+1} i^2 = \sum_{i=1}^{k} i^2 + (k+1)^2 = \{\mathrm{IA}\} = \frac{k(k+1)(2k+1)}{6} + (k+1)^2 =$$

$$= \frac{k(k+1)(2k+1)}{6} + \frac{6(k+1)^2}{6} = \frac{k(k+1)(2k+1) + 6(k+1)^2}{6} = \frac{(k+1)\big((2k^2+k) + (6k+6)\big)}{6} =$$

$$= \frac{(k+1)(2k^2+7k+6)}{6}$$

$$\mathrm{HL} = \frac{(k+1)((k+1)+1)(2(k+1)+1)}{6} = \frac{(k+1)(k+2)(2k+3)}{6} = \frac{(k+1)(2k^2+7k+6)}{6}$$

$$\mathrm{VL} = \mathrm{HL}$$

Då det gäller för $n = 1$ och $n = k + 1$ kommer det därför gälla för alla $n \geq 1$. Att påståendet stämmer följes av induktionsprincipen.

Uppgift 1.2

Bevisa med induktion att:

$$\sum_{j=1}^{n} (2j - 1) = n^2$$

1. Basfall
Jag visar att det gäller för $n = 1$.

$$\text{VL} = 2 \cdot 1 - 1 = 1$$

$$\text{HL} = 1^2 = 1$$

$$\text{VL} = \text{HL}$$

Det stämmer för basfallet $n = 1$.

2. Induktionsantagande
Antag att det stämmer för $n = k$

$$\sum_{j=1}^{k} (2j - 1) = k^2$$

3. Induktionssteg
Visa att det stämmer för $n = k + 1$

$$\sum_{j=1}^{k+1} (2j - 1) = (k + 1)^2$$

$$\text{VL} = \sum_{j=1}^{k+1} (2j - 1) = \sum_{j=1}^{k} (2j - 1) + (2(k + 1) - 1) = \{\text{IA}\} = k^2 + (2k + 2 - 1) = k^2 + 2k + 1$$

$$\text{HL} = (k + 1)^2 = k^2 + 2k + 1$$

$$\text{VL} = \text{HL}$$

Då det gäller för $n = 1$ och $n = k + 1$ kommer det därför gälla för alla $n \geq 1$. Att påståendet stämmer följes av induktionsprincipen.

Uppgift 2

Funktionens syfte är att upphöja x med n enligt namnet på funktionen `expIterative`. Det kan skrivas som x^n. Funktionen utför detta genom en loop som körs n gånger och multiplicerar `res` med x och sparar resultatet i `res`. `res` sätts initiellt till 1 och sedan returnerar funktionen värden av `res`. Detta kan vi skriva som

$$\begin{cases} 1 & \text{om } n = 1 \\ \prod_{i=0}^{n-1} x & \text{om } n \geq 1 \end{cases}$$

Vi kan nu använda induktion för att bevisa funktionens korrekthet, alltså att påståendet nedan stämmer:

$$x^n = \begin{cases} 1 & \text{om } n = 1 \\ \prod_{i=0}^{n-1} x & \text{om } n \geq 1 \end{cases}$$

1. Basfall
Jag visar att det gäller för $n = 0$ och $n = 1$

$$x^0 = 1$$

$$\text{VL} = x^0 = 1$$

$$\mathrm{HL} = 1$$

$$\mathrm{VL} = \mathrm{HL}$$

Det stämmer för basfallet $n = 0$.

$$x^1 = \prod_{i=0}^{0} x$$

$$\mathrm{VL} = x^1 = x$$

$$\mathrm{HL} = x$$

$$\mathrm{VL} = \mathrm{HL}$$

Det stämmer för basfallet $n = 1$.

2. Induktionsantagande
Antag att det stämmer för $n = k$

$$x^k = \prod_{i=0}^{k-1} x$$

3. Induktionssteg
Visa att det stämmer för $n = k + 1$

$$x^{k+1} = \prod_{i=0}^{(k+1)-1} x$$

$$\mathrm{VL} = x^{k+1}$$

$$\mathrm{HL} = \prod_{i=0}^{(k+1)-1} x = \prod_{i=0}^{k} x = \prod_{i=0}^{k-1} x \cdot x = \{\mathrm{IA}\} = x^k \cdot x = x^{k+1}$$

$$\mathrm{VL} = \mathrm{HL}$$

Då det gäller för $n = 0$ och $n = k + 1$ kommer det därför gälla för alla $n \geq 0$. Att påståendet stämmer följes av induktionsprincipen.

Vidare kan vi nu bestämma funktionens tidskomplexitet. Det är väldigt enkelt att se att det endast finns en loop som körs n iterationer. Alltså är tidskomplexiteten $O(n)$

Uppgift 3

Vi ser att denna funktion tar in samma parametrar och ska också exponera ett tal `x`, `n` gånger. För $n \leq 4$ så returnerar den resultatet från `expIterative` . Medan för $n > 4$ så returnerar den ett rekursivt uttryck. Vi kan skriva upp denna funktion som:

$$\mathrm{expRecursive}(x, n) = \begin{cases} \mathrm{expIterative}(x, n) & \text{om } n \leq 4 \\ \mathrm{expRecursive}\left(x, \left\lfloor \frac{n}{2} \right\rfloor\right) \cdot \mathrm{expRecursive}\left(x, \left\lfloor \frac{n+1}{2} \right\rfloor\right) & \text{om } n > 4 \end{cases}$$

Vi kan nu använda induktion för att bevisa att:

$$x^n = \mathrm{expRecursive}(x, n)$$

1. Basfall
Jag visar att det gäller för $n \in \{0, 1, 2, 3, 4\}$

Då blir $\text{expRecursive}(x, n) = \text{expIterative}(x, n)$ där $n \in \{0, 1, 2, 3, 4\}$. Vi har redan bevisat detta enligt induktionsbeviset i uppgift 2. Alltså stämmer basfallet.

2. Induktionsantagande
För $k \in \mathbb{N}$, antag att det gäller för $n \leq k$

$$x^k = \text{expRecursive}(x, k)$$

3. Induktionssteg
Visa att det gäller för $n = k + 1$. Vi börjar med att anta att $k + 1$ är jämn, och att vi kan skriva det som $2m$.

$$k + 1 = 2m$$

$$x^{k+1} = \text{expRecursive}(x, k + 1) \Rightarrow x^{2m} = \text{expRecursive}(x, 2m)$$

$$\text{VL} = x^{2m}$$

$$\text{HL} = \text{expRecursive}(x, 2m) = \text{expRecursive}\left(x, \left\lfloor \frac{2m}{2} \right\rfloor\right) \cdot \text{expRecursive}\left(x, \left\lfloor \frac{2m + 1}{2} \right\rfloor\right) =$$
$$= \text{expRecursive}(x, m) \cdot \text{expRecursive}(x, m) = \{\text{IA}\} = x^m \cdot x^m = x^{2m}$$

$$\text{VL} = \text{HL}$$

Vi har nu visat att det gäller för när $k + 1$ är jämn, nu tar vi för udda.

$$k + 1 = 2m + 1$$

$$x^{k+1} = \text{expRecursive}(x, k + 1) \Rightarrow x^{2m+1} = \text{expRecursive}(x, 2m + 1)$$

$$\text{VL} = x^{2m+1}$$

$$\text{HL} = \text{expRecursive}(x, 2m + 1) = \text{expRecursive}\left(x, \left\lfloor \frac{2m + 1}{2} \right\rfloor\right) \cdot \text{expRecursive}\left(x, \left\lfloor \frac{(2m + 1) + 1}{2} \right\rfloor\right) =$$
$$= \text{expRecursive}\left(x, \left\lfloor \frac{2m + 1}{2} \right\rfloor\right) \cdot \text{expRecursive}\left(x, \left\lfloor \frac{2m + 2}{2} \right\rfloor\right) =$$
$$= \text{expRecursive}(x, m) \cdot \text{expRecursive}(x, m + 1) = \{\text{IA}\} = x^m \cdot x^{m+1} = x^{2m+1}$$

$$\text{VL} = \text{HL}$$

Då det gäller för $n \in \{0, 1, 2, 3, 4\}$ och $n = k + 1$ kommer det därför gälla för alla $n \geq 0$. Att påståendet stämmer följes av induktionsprincipen.

För att räkna ut tidskomplexiteten kan vi använda oss av mästarsatsen. Vi ser att funktionen gör två rekursiva anrop som halverar input-längden och sedan multiplicerar dessa ihop. Vi kan beteckna detta som $T(n) = 2T\left(\frac{n}{2}\right) + O(1)$ där $O(1)$ står för multiplikationen mellan resultatet från de rekursiva anropen. $a = 2, b = 2, c = 0$. $n^{\log_{b(a)}} = n^{\log_2(2)} = n$. $O(1)$ är bara $O(n^0)$, då $0 < 1$ så appliceras den första fallet i mästarsatsen. Tidskomplexiteten blir då $\Theta\left(n^{\log_2(2)}\right) = \Theta(n)$.