Lösningar HW 13

Question 1

Induktion 1: Bevisa följande:

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

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

Solution: (1)
Basfall $n = 1$:

$$\text{V.L: } \sum_{i=1}^{1} i^2 = 1; \quad \text{H.L: } \frac{1(2)(3)}{6} = 1 \quad \text{(OK!)}$$

Induktionsantagande:

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

Induktionssteget:

$$P_{V.L}(n+1) \implies \sum_{i=1}^{n+1} i^2 \iff \sum_{i=1}^{n} i^2 + (n+1)^2 \iff P_{V.L}(n) + (n+1)^2$$

Enligt induktionsantagandet kan vi skriva om $P_{V.L}(n)$ till $\frac{n(n+1)(2n+1)}{6}$, s.a.:

$$P_{V.L}(n) + (n+1)^2 \iff \frac{n(n+1)(2n+1)}{6} + (n+1)^2$$

$$P_{H.L}(n+1) \implies \frac{(n+1)(n+2)(2n+3)}{6} \iff \frac{(n+1)(2n^2+7n+6)}{6} \iff \frac{(n+1)(2n^2+n+6(n+1))}{6}$$

$$P_{H.L}(n+1) \implies \frac{(n+1)(2n^2+n+6(n+1))}{6} \iff \frac{n(n+1)(2n+1)+6(n+1)^2}{6} \iff \frac{n(n+1)(2n+1)}{6} + (n+1)^2$$

Det sista raden implicerar att $P_{H.L}(n+1) = P_{V.L}(n+1)$. Alltså **V.S.B**!

Solution: (2)
Basfall $n = 1$:

$$P_{V.L}(1): \sum_{i=j}^{1} (2j-1) = 1; \quad P_{H.L}(1): \ 1^2 = 1 \quad \text{(OK!)}$$

Induktionsantagande:

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

Induktionssteget:

$$P_{V.L}(n+1) \implies \sum_{j=1}^{n+1} (2j-1) \iff \sum_{j=1}^{n} (2j-1) + 2(n+1) - 1 \iff P_{V.L}(n) + 2n + 1$$

Enligt induktionsantagandet kan vi skriva om $P_{V.L}(n)$ till n^2, s.a.:

$$P_{V.L}(n) + 2n + 1 \iff n^2 + 2n + 1 \iff (n+1)^2 = P_{V.L}(n+1)$$

$$P_{H.L}(n+1) \implies (n+1)^2$$

Alltså $P_{H.L}(n+1) = P_{V.L}(n+1)$ **V.S.B**!

Question 2

Iterativ korrekhet 2

Solution: Koden kan skrivas om som en funktion (loop invarianten efter iterationen j är $\prod_{i=1}^{j} x$):

$$f(x,n) = \prod_{i=1}^{n} x$$

Det vi måste bevisa då är att $f(x,n) = x^n$ som kan göras med hjälp av induktionsbevis:

Basfall $n = 1$:

$$P_{V.L}(1) : x; \quad P_{H.L}(1) : x^1 \quad \text{(OK!)}$$

Induktionsantagande:

$$f(x,n) = x^n$$

Induktionssteget:

$$P_{V.L}(n+1) \implies \prod_{i=1}^{n+1} x \iff x \prod_{i=1}^{n} x \iff xP_{V.L}(n)$$

Enligt induktionsantagandet kan vi ersätta $P_{V.L}(n)$ till x^n, s.a.:

$$P_{V.L}(n+1) \implies xP_{V.L}(n) \iff x \cdot x^n \iff x^{n+1}$$

$$P_{H.L}(n+1) \implies x^{n+1}$$

Alltså $P_{V.L}(n+1) = P_{H.L}(n+1)$. **V.S.B**!

Eftersom $f(x,n)$ är implementerad med hjälp av en for-loop som i sin tur loopar i intervallen $[1,n]$, så är tidskomplexiteten, $\mathcal{O}(f(x,n)) = n$

Question 3

Rekursiv korrekhet: Vi vill bevisa att $P(n+1) = x^{n+1}$

Solution: Eftersom vi har bevisat korrekheten för funktionen 'expIterative' så kan vi börja med att sätta basfallen till $n = 5$:

$$P(5) = P(\frac{n}{2}) \cdot P(\frac{n+1}{2}) \iff P(2) \cdot P(3) = x^2 \cdot x^3 = x^5 \quad \text{(OK!)}$$

Notera att divisionen är heltalsdivision så att i matematiska termer ska $P(n)$ skrivas som:

$$P(n) = P(\lfloor \frac{n}{2} \rfloor) \cdot P(\lfloor \frac{n+1}{2} \rfloor)$$

Vi **antar** att $P(k)$ är korrekt, d.v.s. $P(k) = x^k$ för alla $k \in [1,n] : \mathbb{Z}^+$. Istället för att hålla på med golvfunktionen låt oss expanda golvfunktionen och definiera två versioner av $P(n)$ beroende på om n är udda eller jämn:

$$P(n) = P(\frac{n}{2}) \cdot P(\frac{n}{2}), \quad n = 2k, \; k \in \mathbb{Z}^+$$

$$P(n) = P(\frac{n-1}{2}) \cdot P(\frac{n+1}{2}), \quad n = 2k+1, \; k \in \mathbb{Z}^+$$

Detta kan vi tillämpa för att bevisa att $P(n+1) = x^{n+1}$, alltså att $P(n+1)$ är sann.

Fall 1: $n+1$ är jämn:

$$P(n+1) = P(\frac{n+1}{2}) \cdot P(\frac{n+1}{2}) \iff x^{\frac{n+1}{2}} \cdot x^{\frac{n+1}{2}} = x^{n+1} \tag{3}$$

Fall 2: $n+1$ är udda:

$$P(n+1) = P(\frac{n}{2}) \cdot P(\frac{n+2}{2}) = x^{\frac{n}{2}} \cdot x^{\frac{n+2}{2}} = x^{n+1} \tag{4}$$

Note:-

Ekvivalensen i (3) och (4) gäller eftersom $1 < \frac{n}{2}, \frac{n+1}{2} < n+1$ och för att vi tillämpar induktionsantagandet innan.

Med det så har vi bevisat för både fall att $P(n+1)$ blir x^{n+1} som motsvarar output av funktionen 'expRecursive(x,n+1)'. **V.S.B**!

Vad är **tidskomplexiteten** för den rekursiva funktionen?

Eftersom programmet callar sig själv 2 gånger för en funktions call, så är $a = 2$. Då får varje rekursiv function call, ungefär hälften av datan, blir en input för den rekursiva function call, alltså $b = 2$. Eftersom varje subproblem kommer sluta med att beräkna 'expIterative' för värde i intervallet $[1, 4]$ så kan man säga att kostnaden för varje subproblem blir konstant, alltså $d = 0$. Detta är så eftersom subproblemet behöver **inte** beräkna 'expIterative' med värde i intervallet $[1, k]$ där k beror av n.

Genom att använda mästarsatsen så blir tidskomplexiteten för 'expRecursive' $\Theta(n^{\log_2 2}) \iff \Theta(n)$