DMA 9i

Carl Dybdahl, Patrick Hartvigsen, Emil Chr. Søderblom

January 4, 2017

Part 1

We are given 3 linear homogeneous relation of degree 2 describing the run time of 3 different algorithms.

$$u_n = u_{n-1} + 4 \cdot u_{n-2} \tag{1}$$

$$u_n = 2 \cdot u_{n-1} + 3 \cdot u_{n-2} \tag{2}$$

$$u_n = 9 \cdot u_{n-2} \tag{3}$$

a)

For each algoritm we have been asked to find the characteristic equation and its roots.

The characteristic equation for equation 1 is $x^2 - x - 4 = 0$. To find the roots of this equation we use the quadratic formula $x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$, which yields $x_1 = 1/2 \cdot (1 - \sqrt{17})$, $x_2 = 1/2 \cdot (1 + \sqrt{17})$.

The characteristic equation for equation 2 is $x^2 - 2 \cdot x - 3 = 0$. The roots of this are $x_1 = -1$, $x_2 = 3$.

The characteristic equation for equation 2 is $x^2 - 0 \cdot x - 9 = 0$. The roots of this are $x_1 = 3$, $x_2 = -3$.

b)

The task is to find constants s_1, s_2, s_3 for each algorithm and show the run time is $\Theta(s_1^2)$ for algorithm 1, $\Theta(s_2^2)$ for algorithm 2, $\Theta(s_3^2)$ for algorithm 3.

The a linear homogeneous recurrence relation of degree 2 can be solved using its characteristic equation's roots.

In this case all equations have 2 roots, we can therefore insert the roots in $a_n = u \cdot s_1^n + v \cdot s_2^n$ as s_1 and s_2 and get an function for the recursion equation.

Formula for 1 is:

$$a_n = u \cdot (1/2 \cdot (1 - \sqrt{17}))^n + v \cdot (1/2 \cdot (1 + \sqrt{17}))^n$$

Because u and v is constants we have that $u \cdot s_1^n + v \cdot s_2^n = \Theta(s_1^n + s_2^n)$. s_2 is bigger than s_1 which means that $\Theta(s_1^n + s_2^n) = \Theta(s_2^n)$ and therefore the runtime is

$$\Theta((1/2 \cdot (1 + \sqrt{17})^n)$$

Formula for 2 is:

$$a_n = u \cdot (-1)^n + v \cdot 3^n$$

The same argument as above applyes here and we get.

$$u \cdot (-1)^n + v \cdot 3^n = \Theta((-1)^n + 3^n) = \Theta(3^n)$$

Formula for 3 is:

$$a_n = u \cdot (-3)^n + v \cdot 3^n$$

The same argument as above applyes here and we get.

$$u \cdot 3^n + v \cdot (-3)^n = \Theta 3^n + (-3)^n) = \Theta(3^n)$$

c)

The algorithm with the smallest runtime should go in further development and that is equation 1.

Part 2

a)

We will prove the statement by induction on k. In the base case, $k = 1$, we have to consider two values of n (1 and 2), because $n \leq 2^1 = 2$. When we calculate the statement for $k = 1$ we can see that we have to prove that they are both less than or equal to 13:

$$\begin{aligned} a_n &\leq 3 \cdot 1 \cdot 2^1 + 4 \cdot 2^1 - 1 \\ &= 13 \end{aligned}$$

We can see that they are both less than or equal to 13:

$$a_1 = 3 \leq 13$$

$$\begin{aligned} a_2 &= a_1 + a_1 + 3 \cdot 2 + 1 \\ &= 3 + 3 + 6 + 1 \\ &\leq 13 \end{aligned}$$

Thus we have that $a_1 \leq 13$ and $a_2 \leq 13$, and the statement is therefore true for the base case.

Next we have the inductive case, $k = j + 1$. We assume that $n \leq 2^j \implies a_n \leq 3 \cdot j2^j + 4 \cdot 2^j - 1$, and we must now prove $n \leq 2^{j+1} \implies a_n \leq$

$3 \cdot (j+1) \cdot 2^{j+1} + 4 \cdot 2^{j+1} - 1$. We have that $a_1 = 3$, so the following is for $n \geq 2$:

$$\begin{aligned}
a_n &= a_{\lfloor n/2 \rfloor} + a_{\lceil n/2 \rceil} + 3n + 1 \\
&\leq 3 \cdot j \cdot 2^j + 4 \cdot 2^j - 1 + a_{\lceil n/2 \rceil} + 3n + 1 \\
&\leq 3 \cdot j \cdot 2^j + 4 \cdot 2^j - 1 + 3 \cdot j \cdot 2^j + 4 \cdot 2^j - 1 + 3n + 1 \\
&= 2(3 \cdot j \cdot 2^j + 4 \cdot 2^j - 1) + 3n + 1 \\
&= 6 \cdot j \cdot 2^j + 8 \cdot 2^j - 2 + 3n + 1 \\
&= 6 \cdot j \cdot 2^j + 8 \cdot 2^j + 3n - 1 \\
&= 3 \cdot j \cdot 2^{j+1} + 4 \cdot 2^{j+1} + 3n - 1 \\
&\leq 3 \cdot j \cdot 2^{j+1} + 4 \cdot 2^{j+1} + 3 \cdot 2^{j+1} - 1 \\
&= 3 \cdot (j+1) \cdot 2^{j+1} + 4 \cdot 2^{j+1} - 1
\end{aligned}$$

b)

To see that $a_n = O(n \log n)$, note that $n \leq 2^{\lceil \log_2 n \rceil}$, and thus we have $a_n \leq 3 \cdot \lceil \log_2 n \rceil \cdot 2^{\lceil \log_2 n \rceil} + 4 \cdot 2^{\lceil \log_2 n \rceil} - 1$. As $2^{\lceil \log_2 n \rceil} = \Theta(n)$ and $\lceil \log_2 n \rceil = \Theta(\log n)$, we have $3 \cdot \lceil \log_2 n \rceil \cdot 2^{\lceil \log_2 n \rceil} + 4 \cdot 2^{\lceil \log_2 n \rceil} - 1 = \Theta(\log n \cdot n + 2^{\lceil \log_2 n \rceil}) = \Theta(n \log n)$.