CMPE 300 – Analysis of Algorithms
Fall 2016
Assignment 1 Answers

Question 1 (50 points)

SortDesc algorithm takes an array with n elements and sorts the elements in descending order. Assume n is a positive power of 2.

- Write the complexity of this algorithm T(n) as a recurrence relation. **(15 Points)**
- Solve the recursion and find the exact complexity, then state the complexity in big O notation. **(35 Points)**

```
procedure SortDesc(A[0:n-1]) recursive
input:      A[0:n-1]     (an array of integers with size n)
output:     A[0:n-1]     (array altered by procedure)

if n = 2 then
    tempVar ← -1
    if A[0] < A[1] then
        tempVar ← A[0]
        A[0] ← A[1]
        A[1] ← tempVar
    endif
else
    m ← n/2

    for i ← 0 to m-1 do
        Temp1[i] ← A[i]
        Temp2[i] ← A[i+m]
    endfor

    SortDesc(Temp1)
    SortDesc(Temp2)

    i ← 0
    j ← 0
    k ← 0
    while i < m or j < m
        if i = m
            Temp3[k] ← Temp2[j]
            j ← j + 1
        else if j = m
            Temp3[k] ← Temp1[i]
            i ← i + 1
        else if Temp1[i] < Temp2[j]
            Temp3[k] ← Temp2[j]
            j ← j + 1
        else
```

```
                Temp3[k] ← Temp1[i]
                i ← i + 1
            endif
            k ← k + 1
        endwhile

        for i ← 0 to m-1 do
            A[i] ← Temp3[i]
        endfor
endif
```

Answer 1

```
if n = 2 then
        tempVar ← -1
        if A[0] < A[1] then
                tempVar ← A[0]                 c
                A[0] ← A[1]
                A[1] ← tempVar
        endif
else
        m ← n/2

        for i ← 0 to m-1 do
                Temp1[i] ← A[i]                n
                Temp2[i] ← A[i+m]
        endfor

        SortDesc(Temp1)                        2*T(n/2)
        SortDesc(Temp2)

        i ← 0
        j ← 0
        k ← 0
        while i < m or j < m
                if i = m
                        Temp3[k] ← Temp2[j]
                        j ← j + 1
                else if j = m
                        Temp3[k] ← Temp1[i]
                        i ← i + 1
                else if Temp1[i] < Temp2[j]             n
                        Temp3[k] ← Temp2[j]
                        j ← j + 1
                else
                        Temp3[k] ← Temp1[i]
                        i ← i + 1
                endif
                k ← k + 1
        endwhile

        for i ← 0 to m-1 do
                A[i] ← Temp3[i]                n
        endfor
endif
```

$$T(n) = 2 * T\left(\frac{n}{2}\right) + 3n$$

$$= 2 * \left(2 * T\left(\frac{n}{4}\right) + \frac{3}{2}n)\right) + 3n$$

$$= 4 * T\left(\frac{n}{4}\right) + 6n$$

$$= 4 * \left(2 * T\left(\frac{n}{8}\right) + \frac{3}{4}n)\right) + 6n$$

$$= 8 * T\left(\frac{n}{8}\right) + 9n$$

Say, $n = 2 * 2^k$ and $k = \log_2\left(\frac{n}{2}\right)$

$$T(n) = 2^k * T\left(\frac{n}{2^k}\right) + 3kn$$

$$= \frac{n}{2} * T(2) + 3 * \log_2\left(\frac{n}{2}\right) * n$$

Say , $T(2) = C$

$$T(n) = \frac{n}{2} * C + 3 * n * \log_2\left(\frac{n}{2}\right) \in O(n * \log n)$$

Question 2 (50 Points)

Consider the given function $f(n)$ and determine whether the following cases are true or false. Justify your answers formally ie. show all your work in deriving your answer. (Hint: Use Stirling's Approximation)

$$f(n) = n^4 + n * log(5n! * n^4) + n^2\sqrt{n} + 42$$

1. $f(n) \in O(n^6)$ **(10 Points)**
2. $f(n) \in o(n^5)$ **(15 Points)**
3. $f(n) \in \Omega(n^3 * log(n))$ **(10 Points)**
4. $f(n) \in \Theta(n^5 * log(n))$ **(15 Points)**

Answer 2

We use Stirling's approximation to simplify the term $log(n!)$.

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

The function f(n) becomes

$$\begin{aligned}
f(n) &= n^4 + n^4 * log(5n! * n^4) + n^2\sqrt{n} + 42 \\
&= n^4 + n^4 * log(5) + n^4 * log(n!) + n^4 * log(n^4) + n^2\sqrt{n} + 42 \\
&= n^4 + n^4 * log(5) + n^4 * log(n!) + 4n^4 * log(n) + n^2\sqrt{n} + 42 \\
&\approx n^4 + n^4 * log(5) + n^4 * log(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n) + 4n^4 * log(n) + n^2\sqrt{n} + 42 \\
&= n^4 + n^4 * log(5) + n^4 * log(\sqrt{2\pi n}) + n^4 * log(\left(\frac{n}{e}\right)^n) + 4n^4 * log(n) + n^2\sqrt{n} + 42 \\
&= n^4 + n^4 * log(5) + n^4 * log(\sqrt{2\pi n}) + n^5 * log\left(\frac{n}{e}\right) + 4n^4 * log(n) + n^2\sqrt{n} + 42 \\
&= n^4 + n^4 * log(5) + n^4 * log(\sqrt{2\pi n}) + n^5 * log(n) - n^5 * log(e) + 4n^4 * log(n) + n^2\sqrt{n} + 42
\end{aligned}$$

Therefore

$$\begin{aligned}
&f(n) = n^5 * log(n) - n^5 * log(e) + O(n^4 * log(n)) \\
&f(n) \in O(n^5 * log(n))
\end{aligned}$$

1. $f(n) \in O(n^6)$: True

   If we can find a c and n_0 such that $f(n) \leq c * g(n), \forall n \geq n_0$, then $f(n) \in O(g(n))$. Consider the term $n^5 * (log(n) - log(e))$:

   $n^5 * (log(n) - log(e)) \leq c * n^6$

   $\frac{log(n) - log(e)}{n} \leq c$

   Let $c = 1$ for all values of $n \geq 1$ ($n_0 = 1$). This implies that $f(n) \leq c * g(n)$.

2. $f(n) \in o(n^5)$: False

   We can directly show that there is no c and no n_0 such that $f(n) \leq n^5$. Let's look at the term $n^5 * log(n) - n^5 * log(e)$.

   $n^5 * (log(n) - log(e)) \leq c * n^5, \forall n \geq n_0$

   $n * (log(n) - log(e)) \leq c$

   $n * log(n)$ is a monotonically increasing function and c is constant. For all $\{c, n_0\}$ pairs, there is an n value, which is greater than n_0, which makes $n * (log(n) - log(e)) > c$. We don't need to look at other terms.

3. $f(n) \in \Omega(n^3 * log(n))$: True

   If we can find a c and n_0 such that $f(n) \geq c * g(n), \forall n \geq n_0$, then $f(n) \in \Omega(n^2 * log(n))$. Consider the term $n^5 * (log(n) - log(e))$:

   $c * n^3 * log(n) \leq n^5 * (log(n) - log(e))$

   $c \leq n^2 * \frac{log(n) - log(e)}{log(n)}$

   Let $c = 1$ for all values of $n \geq 3$ $(n_0 = 3)$. This implies that $c * g(n) \leq f(n)$ since the remaining terms of $f(n)$ are positive.

4. $f(n) \in \Theta(n^5 * log(n))$: True

   $\lim_{n \to \infty} \frac{f(n)}{n^5 * log(n)} = 1 = c$

   Therefore $f(n) \in \Theta(n^5 * log(n))$