CSE 101 Homework 0 Solutions

Winter 2021

Question 1 (Program Runtimes, 20 points). *Consider the following programs:*

```
Alg1(n):
For i = 1 to n
   j = 1
   while j < n
      j = j+1

Alg2(n):
For i = 1 to n
   j = 1
   while j < n
      j = j+i
```

For each of these algorithms, compute the asymptotic runtime in the form $\Theta(-)$*.*

Solution 1.

Alg1: Each iteration of the `for` loop involves $\Theta(n)$ operations, as j is always incremented $n-1$ times. There are n iterations of the `for` loop, so our total runtime is $n * \Theta(n) = \boxed{\Theta(n^2)}$.
Alg2: In each iteration of the `for` loop, j is incremented $\Theta(\frac{n}{i})$ times. i iterates from 1 to n, so our total runtime is $\Theta(n)+\Theta(\frac{n}{2})+\Theta(\frac{n}{3})+\cdots = \sum_{i=1}^{n} \Theta(\frac{n}{i}) = n \cdot \sum_{i=1}^{n} \Theta(\frac{1}{i})$. The summation is just the n^{th} harmonic number H_n, and $H_n = \Theta(\log n)$, so our total runtime is $\boxed{\Theta(n \log n)}$.

Question 2 (Big-O Computations, 20 points). *Sort the functions below in terms of their asymptotic growth rates. Which of these functions have polynomial growth rates? Remember to justify your answers.*

- $a(n) = n + n^{1/2} + n^{1/3} + \ldots + n^{1/n}$
- $b(n) = 3^{\lceil \log_2(n) \rceil}$
- $c(n) = n^2(2 + \cos(n))$
- $d(n) = n^{100} 2^{n/2}$
- $e(n) = 2^n$

Solution 2.

- $n \leq a(n) \leq n + \log(n)n^{1/2} + n \times n^{1/\log(n)} = O(n)$ since $\log(n) \in O(n^{1/2})$. Notice that $l = n^{1/\log(n)}$ is a constant since $\log(l) = 1/\log(n) * \log(n) = 1 \implies l = e$. So $a(n) \in \theta(n)$.
- $b(n) = 3^{\lceil \log_2(n) \rceil} \leq 3 \times 3^{\log_2(n)} = 3 \times 2^{\log_2(3)\log_2(n)} = 3n^{\log_2(3)}$ and $b(n) = 3^{\lceil \log_2(n) \rceil} \geq 3^{\log_2(n)}/3 = 2^{\log_2(3)\log_2(n)}/3 = n^{\log_2(3)}/3$. So $b(n) = \theta(n^{\log_2(3)})$.
- $\Omega(n^2) \leq c(n) = n^2(2 + \cos(n)) \leq O(n^2)$ since $1 \leq 2 + cos(n) \leq 3$.
- $\Omega(n^2) \leq d(n) = n^{100} 2^{n/2}$

- $\Omega(n^{100}2^{n/2}) \leq e(n) = 2^n$ since $n^{100} \in O(2^{n/2})$

So the order is clearly $\boxed{a, b, c, d, e}$ by reasons above. The functions a, b, c are polynomial growth rates. The functions d, e are not polynomial growth since they are greater than $2^{n/2}$ which is not polynomial growth.

Question 3 (Walks and Paths, 30 points)**.** *In a graph G we say that there is a* walk *from vertex u to another vertex w if there is a sequence of vertices $u = v_0, v_1, \ldots, v_n = w$ so that (v_i, v_{i+1}) is an edge of G for each $0 \leq i < n$. Prove that if there is a walk from u to w there is a walk where all of the vertices v_i are distinct. Hint: if two are the same show how you can use this to construct a shorter walk.*

Solution 3.

Suppose we have a walk from vertex u to vertex w, the walk is denoted as $(u, v_0, v_1, ..., v_n, w) = p$ where there are duplicated vertices in p. Assume that we cannot find a walk $\hat{p}$ such that $\hat{p} \subset p$ and elements in $\hat{p}$ are distinct. Suppose $v_i, v_j \in p, v_i = v_j, i < j$. Then (v_i, v_{j+1}) is also an edge of G. So $\hat{p} = (u, v_0, ..., v_i, v_{j+1}, ..., v_n, w)$ is a valid walk. We can apply this procedure until all the elements are distinct. This contradicts with the assumption we have. So there exists a walk from v to w where all the vertices v_i are distinct.

Question 4 (Recurrence Relations, 30 points)**.** *Consider the recurrence relation*

$$T(1) = 1, \quad T(n) = 2T(\lfloor n/2 \rfloor) + n.$$

(a) What is the exact value of $T(2^n)$?

(b) Give a Θ expression for $T(n)$. Hint: compare its value to that at nearby powers of 2.

(c) Consider the following purported proof that $T(n) = O(n)$ by induction:

If $n = 1$, then $T(1) = 1 = O(1)$.

If $T(m) = O(m)$ for $m < n$, then

$$T(n) = 2T(\lfloor n/2 \rfloor) + n = O(n) + O(n) = O(n).$$

Thus, $T(n) = O(n)$.

What is wrong with this proof? Hint: consider the implied constants in the big-Os.

Solution 4.

(a) We show by induction that $\forall n \geq 0$, $T(2^n) = (n + 1) \cdot 2^n$. For $n = 1$, clearly $T(2^0) = 1 = (0 + 1) \cdot 1$. Suppose the relationship holds for all $n \leq t$. Then, $T(2^{n+1}) = 2T(2^n) + 2^{n+1} = 2(n + 1) \cdot 2^n + 2^{n+1} = \boxed{(n + 2) \cdot 2^{n+1}}$.

(b) Let $k = \lfloor \log_2 n \rfloor$ Observe that $T(2^k) \leq T(n) \leq T(2^{k+1})$ or

$$(\lfloor \log_2 n \rfloor + 1)2^{\lfloor \log_2 n \rfloor} \leq T(n) \leq (\lfloor \log_2 n \rfloor + 2)2^{\lfloor \log_2 n \rfloor + 1}$$

implying

$$\log_2 n \cdot n/2 \leq T(n) \leq 2(\log_2 n + 2)n.$$

Thus $\boxed{T(n) = \Theta(n \log_2 n)}$.

(c) In order for the induction step to be correct, we require that the same constant as that of $T(m)$ for $m < n$ is also obtained for $T(n)$. Under the assumption that $\forall m < n$, $T(m) = O(m)$, let c be a constant such that for a sufficiently large m, $T(m) \leq c \cdot m$. This only implies that $T(n) = 2T(\lfloor n/2 \rfloor) + n \leq (c+1)n$ and the induction step fails.