CSE 101 Homework 0

Winter 2022

This homework is due on gradescope Friday January 7th at 11:59pm on gradescope. Remember to justify your work even if the problem does not explicitly say so. Writing your solutions in \LaTeX is recommend though not required.

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

Alg1(n)
\begin{verbatim}
for i = 1 to n
   for j = 1 to 2^n
      Print(j)
\end{verbatim}

and

Alg2(n)
\begin{verbatim}
for i = 1 to n
   for j = 1 to 2^i
      Print(j)
\end{verbatim}

For each of these programs give the asymptotic runtime as $\Theta(f(n))$ for some function $f$ and justify your work.

**Solution 1.**

Alg1(n): Each iteration of the for loop involves $\Theta(2^n)$ as $j$ is incremented $2^n - 1$ times. And there are $n$ iterations of the for loop, so the total runtime is $n \cdot \Theta(2^n) = \Theta(n \cdot 2^n)$.

Alg2(n): In each iteration of the inner for loop, $j$ runs $2^i$ times. And we know that $i$ iterates from 1 to $n$. Thus, we have a geometric progression of the form $2^1 + 2^2 + \ldots + 2^n$. Therefore the total runtime is $\Theta(2^n)$.

**Question 2** (Asymptotic Comparisons, 20 points). Sort the following functions of $n$ in terms of their asymptotic growth rates. In particular, ones should go later in the list if they are larger when sufficiently large values of $n$ are used as inputs. Which of these functions have polynomial growth rates? Remember to justify your answers.

- $a(n) = (\lfloor \log_2(n) \rfloor)!$
- $b(n) = n(n + 100)(n + 10000)$
- $c(n) = e^{\sqrt{n}}$
- $d(n) = 6^{\log_2(n)}$
- $e(n) = 10^{10^{10}} n^2$

**Solution 2.**

For $b(n)$, on opening the brackets we would get $O(n^3)$ as the overall time complexity. We can ignore the other $n$ terms as they would be smaller than $n^3$. 

For \( d(n) \), we can rewrite \( 6^{\log_2 n} \) as \( 2^{\log_2 6 \cdot \log_2 n} \), or \( (2^{\log_2 n})^\log_2 6 \). Thus we end up with \( n^{(\log_2 6)} \).

Since \( 3 > \log_2 6 > 2 \), we know \( b(n) \) will have a time complexity less than \( O(n^3) \) and greater than \( O(n^2) \).

For \( e(n) \), we can simply ignore the constant term, and that gives a time complexity of \( O(n^2) \).

Thus, \( b(n), d(n) \) and \( e(n) \) have polynomial growth rates, and we know \( e(n) < d(n) < b(n) \).

Since \( c(n) \) is an exponential function, we know that it will grow faster than any polynomial function. However, it is nontrivial to show this for \( a(n) \). We can see that \( a(n) \) has superpolynomial growth by substituting \( n = 2^{n'} \) into an arbitrary \( n^k \):

\[
\begin{align*}
(|\log_2(n)|)! & \geq n^k \\
(|\log_2(2^{n'})|)! & \geq (2^{n'})^k \\
n! & \geq 2^{kn'}
\end{align*}
\]

The left side is then greater than the right side for some \( n \). To see this, note that

\[
n! = 1 \cdot 2 \cdot \ldots \cdot n
\]

and

\[
2^n = 2 \cdot 2 \cdot \ldots \cdot 2
\]

(where both have the same number of terms, \( n \)). Since we can choose an \( n \) such that the factorial has much larger terms, the factorial eventually beats the exponential. One method for proving this formally is to use the fact that the second half of the product in the factorial is greater than \( n/2 \):

\[
n! > (n/2)^{n/2} > 2^{n \log(n/2)/2} > 2^{nk}
\]

One alternative way to show that the factorial beats the exponential is to use Stirling’s approximation, one version of which says that

\[
\ln(n!) = n\ln(n) - n + \Theta(\ln n)
\]

since \( \ln(2^n) < \ln(e^n) = O(n) \) and

\[
\ln(n!) = O(n\log(n))
\]

, it is clear that \( n! \gg 2^n \) since \( O(n\log(n)) \gg O(n) \). Thus, \( a(n) \) is superpolynomial.

We now compare \( a(n) \) and \( c(n) \). We perform a similar substitution as in the previous paragraph, substituting \( n - > 2^{n'} \). This gives

\[
a(n') = n'!
\]

and

\[
c(n') = e^{2^{n'/2}}
\]

Now, we must compare the factorial to the double exponential. We note that by comparing the terms in their respective products, it is clear that \( n! < n^n \). This fact allows us to compare \( a(n) \) and \( c(n) \) more easily:

\[
n'! < n'n'
\]

\[
= (e^{\log(n')})^{n'}
\]

\[
= e^{n' \log(n')} < e^{2^{n'/2}}
\]

And thus \( a(n) < c(n) \) (for some \( n \)).

Hence, the order in terms of their asymptotic growth rates is:  
\[
e(n) < d(n) < b(n) < a(n) < c(n)
\]
Question 3 (Extremal Graph Theory, 30 points). Say that a graph $G$ has a path of length three if there exist distinct vertices $u, v, w, t$ with edges $(u, v), (v, w), (w, t)$. Show that a graph $G$ with 99 vertices and no path of length three has at most 99 edges.

Solution 3.

We will show that a graph $G$ with 99 vertices and no path of length of three has at most 99 edges. We actually show a more general case: that any graph with at most $n$ vertices and no path of length 3 has at most $n$ edges.

One key technique for solving this problem (and this is a useful problem-solving technique in general) is to play with some small example graphs and notice the patterns that tend to appear in the connected components – namely, that all of the components tend to manifest as star graphs (one vertex in the center connected to a ton of other vertices), paths of length $\leq 2$, or triangle graphs (note that the triangle graphs do not break the “path of length 3” condition, since the 3 vertices are not distinct). This taxonomy of connected components naturally suggests a proof by cases.

In the proof by cases, we examine each connected component individually. Below, $|V_i|$ denotes the number of vertices in connected component $i$, and $|E_i|$ denotes the number of edges in that connected component. In each connected component, we can look at the vertex $v$ of max degree $d$, since all cases (besides the triangle graph), the other vertices in the connected component must be connected to $v$ and nothing else (or else there would be a path of length 3). $v$ must satisfy one of the following cases:

- $d = 0$ In this case, $v$ is not connected to anything. Thus this connected component has 0 edges and 1 vertex, and $|E_i| \leq |V_i|$.
- $d = 1$ In this case, $v$ is connected to a single other vertex $u$. $u$ cannot be connected to any other vertex, as the degree is at most 1. Then $|E_i| = 1$ and $|V_i| = 2$, so $|E_i| \leq |V_i|$.
- $d = 2$ Then $v$ is connected to $u$ and $w$. If $u$ or $w$ are connected to any other vertices (other than each other), there is a path of length 3. This is true because we can label one of these extra vertices $t$ and form a path $t - u - v - w$ (or $t - w - v - u$). In either of the allowed cases, $|E_i| \leq |V_i|$.
- $d \geq 3$ Then $v$ is connected to $d$ other vertices. If any of these other vertices have neighbors besides $v$, we can form a path of length 3. More formally, if some vertex $u$ connected to $v$ has degree $> 1$, it is connected to some other vertex $w$ in the connected component. Additionally, since $d \geq 3$, we can choose another vertex $t \neq u, w$ connected to $v$. We can then form a path $w - u - v - t$. Therefore the only allowed form for this connected component is the star graph (one vertex of degree $d$ and the rest of degree 1), which has $|E_i| = |V_i| - 1 \leq |V_i|$.

So in each of the connected components, we have $|E_i| \leq |V_i|$. Since a graph’s edge/vertex counts can be decomposed as the sums of the corresponding quantities in its connected components, we can write

$$|E| = \sum_i |E_i| \leq \sum_i |V_i| = |V|$$

and so a graph that satisfies the given properties must always have at most $|V|$ edges.

Question 4 (Recurrence Relation, 30 points). Suppose that you have a function $T(n)$ defined by $T(1) = 1$ and

$$T(n) = T(n - 1) + T(\lfloor n/2 \rfloor)$$

for $n > 1$.

(a) Show that for any positive integer $k$ that $T(n)$ grows faster than $n^k$. Hint: Show that $T(n) \geq T(n/4)$. [15 points]

(b) Consider the following “proof” that $T(n) = O(1)$ (note that this contradicts part (a)):

We proceed by strong induction on $n$. Clearly $T(1) = O(1)$, which gives us our base case. If we assume that $T(m) = O(1)$ for all $m < n$, then $T(n) = T(n - 1) + T(\lfloor n/2 \rfloor) = 2O(1) = O(1)$. This completes our inductive step and proves that $T(n) = O(1)$ for all $n$.

What is wrong with the above proof? (Hint: Consider what the implied constant in the $O$ term would be.) [15 points]
Solution 4. (a) We can begin by showing that the hint is true. We know that

\[ T(n) = T(n - 1) + T([n/2]) \]

We can expand this recursive formula into a sum:

\[ T(n) = \sum_{i=1}^{n} T([i/2]) \]

The hint should now appear closer, since the last half of this sum is composed entirely of terms \( \geq T([n/4]) \). This is proven more formally by splitting the sum into two parts:

\[
T(n) = \sum_{i=1}^{n/2} T([i/2]) + \sum_{j=n/2}^{n} T([j/2]) \\
\geq \sum_{i=1}^{n/2} T([i/2]) + \sum_{j=n/2}^{n} T([n/4]) \\
\geq \left( \sum_{i=1}^{n/2} T([i/2]) \right) + [n/2]T([n/4]) \\
\geq [n/2]T([n/4])
\]

Next, we must show that this expression implies that \( T(n) \) grows faster than \( n^k \). Given the hint, we can show that

\[ T(n) \geq \left( \frac{n}{2} \right) \cdot \left( \frac{n}{8} \right) \cdot \left( \frac{n}{32} \right) \cdots \frac{n}{2^k} \]

\[ = \frac{n^k}{C_k} \]

where \( C_k \) is some fixed constant based on \( k \). Thus for any \( k \), we can choose a value of \( n \) such that the product above becomes larger than \( n^k \), meaning that \( T(n) \) grows faster than \( n^k \).

(b) The assumption \( T(m) = O(1) \) really means that there is a constant \( C \) such that \( T(m) < C \) for all \( m \). If we assume that this is true for all \( m < n \), the recurrence relation becomes

\[ T(n) = T(n - 1) + T([n/2]) \leq C + C = 2C \]

But this does not match the inductive hypothesis that we started with. The constant changed from \( C \) to \( 2C \). Hence, the proof is incorrect.

Question 5 (Extra credit, 1 point). Approximately how much time did you spend working on this homework?