CS341 Algorithms

Assignment 1

Reuben Drummond

1. (a) [9 marks] Prove or disprove each of the following statements.

   i. if $\log(f(n)) \in \Omega(\log(g(n)))$ then $f(n) \in \Omega(g(n))$.

   Consider functions for $k \in \mathbb{N}$,

$$\begin{aligned} f(n) &= n^k \\ g(n) &= n^{k+1} \end{aligned}$$

   We have that,

$$\begin{aligned} \log f(n) &= kn \\ \log g(n) &= (k+1)\,n \end{aligned}$$

   It is true that $\log f(n) \in \Omega\left(g\left(n\right)\right)$ since,

$$\exists c, n_0 > 0 : \forall n > n_0, c \log g\left(n\right) \leq \log f\left(n\right) \tag{1}$$

   And we can select a $c \forall n_0$ which satisfies the definition,

$$\begin{aligned} c \log g\left(n\right) &\leq \log f\left(n\right) \\ ckn &\leq \left(k+1\right) n \\ c &\leq \frac{\left(k+1\right)}{k} \quad \text{(since } n > 0\text{)} \end{aligned}$$

   Assume that $f\left(n\right) \in \Omega g\left(n\right)$. By **??**, we can see that,

$$\begin{aligned} cg\left(n\right) &\leq f\left(n\right) \\ cn^{k+1} &\leq n^k \\ c &\leq \frac{n^{k+1}}{n^k} = n \end{aligned}$$

   Clearly, we cannot select a finite value of c for which the condition for 1 hold, which is a contradiction to the claim that $f\left(n\right) \in \Omega g\left(n\right)$.

   Since we have found a counter example we can disprove the original statement.

   ii. If $f(n) \in O(h(n))$ and $g(n) \in O(h(n))$ then $\dfrac{f(n)}{g(n)} \in O(1)$.

   Consider the functions,

$$\begin{aligned} f\left(n\right) &= n \\ g\left(n\right) &= 1 \\ h\left(n\right) &= n^2 \end{aligned}$$

Clearly, $f(n), g(n) \in h(n)$ since,

$$\exists c_1, n_0 : \forall n > n_0, f(n) \leq c_1 h(n)$$
$$\exists c_2, n_0 : \forall n > n_0, g(n) \leq c_2 h(n)$$

Now consider,

$$\frac{f(n)}{g(n)} = \frac{n}{1} = n \notin O1$$

We can therefore disprove the original statement by counter example.

iii. If $f(n) \in o(g(n))$ then $\log(f(n)) \in o(\log(g(n)))$.
Consider functions such that $f(n) \in o(g(n))$,

$$f(n) = n$$
$$g(n) = n^2$$

If $\log f(n) \in o(\log g(n))$ we must have $\forall c > 0$,

$$\begin{aligned}
\log f(n) &\leq c \log g(n) \\
\log n &\leq c \log n^2 \\
\log n &\leq c2 \log n \\
1 &\leq c2 \quad \text{since } n > 0 \\
\frac{1}{2} &\leq c \\
c &\geq \frac{1}{2}
\end{aligned}$$

Since $c \geq \frac{1}{2}$, we conclude that $\log f(n) \notin o(\log g(n))$. Therefore, we have disproven the original statement by counter example.

(b) [6 marks] Analyze the following pseudocode and give a tight (Θ) bound on the running time as a function of n. You can assume that all individual instructions (including logarithm) are elementary, i.e., take constant time. Show your work.

```
l := 0;
for i = n+1 to n^2 do
  for j = 1 to ⌈log i⌉ do
    l := l+1
  od
od.
```

There are two loops which can be expressed as a nested sum. Note that the execution time of operations are denoted as c_1 through to c_4 in order. Expressed as a sum, we have that the running time is,

$$
\begin{aligned}
\text{running time} &= c_1 + \sum_{i=n+1}^{n^2} \left(c_2 + \sum_{j=0}^{\log i} (c_3 + c_4) \right) \\
&= c_1 + \sum_{i=n+1}^{n^2} (c_2 + (c_3 + c_4) \log i)
\end{aligned}
$$

We can say that,

$$
\sum_{i=n+1}^{n^2} (c_2 + (c_3 + c_4) \log i) \leq \int_{x}^{x^2} (c_2 + (c_3 + c_4) \log x) \, dx
$$

Since $\log x$ is monotonically increasing and the sum can be viewed as a Riemann sum of width 1.

Noting that,

$$
\int \log_b x dx = \frac{1}{\ln b} (x \ln x - x) + c
$$

we have,

$$
\begin{aligned}
\text{running time} &\leq c_1 + \int_{n}^{n^2} (c_2 + (c_3 + c_4) \log_b x) \, dx \\
&\leq c_1 + \left[c_2 x + (c_3 + c_4) \frac{1}{\ln b} (x \ln x - x) \right]_{n}^{n^2} \\
&= c_1 + c_2 \left(n^2 - n \right) + (c_3 + c_4) \frac{1}{\ln b} \left[(n^2 \ln n^2 - n^2) - (n \ln n - n) \right]
\end{aligned}
$$

Clearly, the highest order term is proportional to $n^2 \log n$. Therefore, we can conclude the tight bound is $\Theta \left(n^2 \log n \right)$.

Appendices