College of Computer & Information Science	CS5800
Northeastern University	MS Algorithms
Fall 2017	23 September 2017

Sample Solutions to Problem Set 1

1. (6 points) Instability of a local improvement algorithm

As we discussed in class, there may be several approaches to finding a stable matching. One reasonable approach is the following. Suppose we start with an arbitrary matching, and then repeat the following step until there are no unstable pairs.

- If there exists unstable pairs (m, w) and (m', w') such that m prefers w' over w and w' prefers m over m', then replace the pairs by (m, w') and (m', w).

The above local improvement algorithm is fairly natural. But it does not work! Cycles may occur, especially if you choose the "wrong" unstable pairs to swap, causing the algorithm to loop forever. Consider the following preference lists with 3 women A B, C, and 3 men U, V, and W.

A	B	C	U	V	W
U	W	U	B	A	A
W	U	V	A	B	B
V	V	W	C	C	C

For the above preference list, show that there exists a 4-step cycle in the local improvement algorithm, starting with the matching $\{(A, U), (B, V), (C, W)\}$.

Answer:

$$\begin{array}{r} (A,U)(B,V)(C,W) \\ B \leftrightarrow U \\ (A,V)(B,U)(C,W) \\ B \leftrightarrow W \\ (A,V)(B,W)(C,U) \\ A \leftrightarrow W \\ (A,W)(B,V)(C,U) \\ B \leftrightarrow U \\ (A,U)(B,V)(C,W) \end{array}$$

2. (6 points) Stability in competition Chapter 1, Exercise 3, page 22.

Answer: There is not always a stable pair of schedules. Suppose Network A has two shows a_1 and a_2 with ratings 2 and 4, respectively, while Network B has two shows b_1 and b_2 with ratings 1 and 3, respectively. Each network has two possible schedules. Network A would prefer a_1 to be in the same slot as b_1 and a_2 in the same slot as b_2 so that A can win both slots. On the other hand, network B prefers b_1 to have the same slot as a_2 and b_2 to have the same slot as a_1 so that B wins at least one slot. So regardless of the pair of schedules chosen, one of A or B would like to switch the order.

3. (6 points) Ordering functions

Arrange the following functions in order from the slowest growing function to the fastest growing function. Briefly justify your answers. (*Hint:* It may help to plot the functions and obtain an estimate of their relative growth rates. In some cases, it may also help to express the functions as a power of 2 and then compare.)

$$\sqrt{n} \quad n\sqrt{\lg n} \quad 2^{\sqrt{\lg n}} \quad (\lg n)^2$$

Answer: The order from slowest growing to the fastest growing function is: $(\lg n)^2$, $2^{\sqrt{\lg n}}$, $\sqrt{n}$, $n\sqrt{\lg n}$. The informal reasoning behind this is as follows. The function $(\lg n)^2$ clearly grows slowly than $n\sqrt{\lg n}$. Where do we place $2^{\sqrt{\lg n}}$ and $\sqrt{n}$? The function $\sqrt{n}$ is a polynomial and grows faster than $(\lg n)^2$ and clearly slower than $n\sqrt{\lg n}$ since the exponent on n is $1/2 < 1$.

That leaves $2^{\sqrt{\lg n}}$. The function $\lg n$ can be written as $2^{\lg\lg n}$ and $\sqrt{n}$ as $2^{(\lg n)/2}$. Comparing the three functions $2^{\sqrt{\lg n}}$, $\lg n$, and $\sqrt{n}$ is equivalent to comparing $\sqrt{\lg n}$, $\lg\lg n$ and $(\lg n)/2$. Since $\sqrt{\lg n}$ is greater than $\lg\lg n$ and smaller than $(\lg n)/2$, we can place $2^{\sqrt{\lg n}}$ between the other two functions.

We now present a formal proof using three steps: (i) $(\lg n)^2 = o(2^{\sqrt{\lg n}})$, (ii) $2^{\sqrt{\lg n}} = o(\sqrt{n})$, and (iii) $\sqrt{n} = o(n\sqrt{\lg n})$.

(i) We use the connection between asymptotic notation and limits.

$$\begin{aligned}
\lim_{n\to\infty} \frac{(\lg n)^2}{2^{\sqrt{\lg n}}} &= \lim_{m\to\infty} \frac{m^2}{2^{\sqrt{m}}} \\
&= \lim_{m\to\infty} \frac{2m}{2^{\sqrt{m}} \ln 2/(2\sqrt{m})} \\
&= \lim_{m\to\infty} \frac{4m^{3/2}}{2^{\sqrt{m}}} \\
&= \lim_{m\to\infty} \frac{6\sqrt{m})}{2^{\sqrt{m}}(\ln 2)^2/(2\sqrt{m})} \\
&= \lim_{m\to\infty} \frac{12m}{2^{\sqrt{m}}(\ln 2)^2} \\
&= \lim_{m\to\infty} \frac{12}{2^{\sqrt{m}}(\ln 2)^3/(2\sqrt{m})} \\
&= \lim_{m\to\infty} \frac{24\sqrt{m}}{2^{\sqrt{m}}(\ln 2)^3} \\
&= \lim_{m\to\infty} \frac{24/(2\sqrt{m})}{2^{\sqrt{m}}(\ln 2)^4/(2\sqrt{m})} \\
&= \lim_{m\to\infty} \frac{24}{2^{\sqrt{m}}(\ln 2)^4}
\end{aligned}$$

$$= \quad 0,$$

where we replace $\lg n$ by m in the first step (since $\lg n$ is monotonically increasing in n) and use L'Hopital's rule multiple times. The final step holds since $2^{\sqrt{m}}$ goes to infinity as m goes to infinity. Therefore, $(\lg n)^2 = o(2^{\sqrt{\lg n}})$.

(ii) We again use the connection between asymptotic notation and limits.

$$\begin{aligned}
\lim_{n\to\infty} \frac{2^{\sqrt{\lg n}}}{\sqrt{n}} &= \lim_{n\to\infty} \frac{2^{\sqrt{\lg n}}}{2^{(\lg n)/2}} \\
&= \lim_{n\to\infty} \frac{1}{2^{(\lg n)/2-\sqrt{\lg n}}} \\
&= 0,
\end{aligned}$$

since $(\lg n)/2 - \sqrt{\lg n}$ goes to ∞ as n goes to ∞. Therefore, $2^{\sqrt{\lg n}} = o(\sqrt{n})$.

(iii) We again use the connection between asymptotic notation and limits.

$$\begin{aligned}
\lim_{n\to\infty} \frac{\sqrt{n}}{n\sqrt{\lg n}} &= \lim_{n\to\infty} \frac{1}{\sqrt{n}(\lg n)} \\
&= 0,
\end{aligned}$$

Therefore, $\sqrt{n} = o(n\sqrt{\lg n})$.

4. (6 points) Chapter 2, Exercise 6, page 68.

Answer:

(a) The outer loop runs for n iterations. The inner loop runs for at most n iterations. In a single iteration of the inner loop, the addition takes at most n steps, and the storage at most 1 step. So the total number of steps is at most $n \cdot n \cdot (n+1) = n^3 + n^2$, which is $O(n^3)$. Note that this is an upper bound only.

(b) Using the same logic as above, we can say the outer loop runs for n iterations, the inner loop runs for at least 1 iteration, and the addition and storage take at least 2 steps. So the total time is at least $n \cdot 1 \cdot 2 = \Omega(n)$. This is much too weak a bound since there is a big gap between the upper and lower bounds. So we need to strengthen one of the two.

Let us consider the inner loop again. It runs for $n-1$ iterations the first time, $n-2$ the second time, and so on. So during at least half of the iterations of the outer loop, the inner loop runs at least $n/2$ iterations. Furthermore, the number of additions done inside an inner loop is also quite often as high as $n/2$. So this suggests that $\Omega(n^3)$ is more likely to be a lower bound as well.

Let us make a more precise calculation. Using the ranges of the variables i and j, the total number of steps can be written as

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} (j-i+1)$$

($j-i$ for the additions and 1 for the storage). Let us focus on the inner sum first. For $i=1$, it is $2+3+\ldots+n$. For $i=2$, it is $3+4+\ldots+n$. So for general i, it is $(i+1)+(i+2)+\ldots+n$. Using the arithmetic progression formula or noting that this is the sum of the first n numbers minus the sum of the first i numbers, we get $n(n+1)/2 - i(i+1)/2$.

So our desired sum is now

$$\begin{aligned}
\sum_{i=1}^{n}\left(\frac{n^2+n}{2}-\frac{i^2+i}{2}\right) &= \frac{n^3+n^2}{2}-\frac{1}{2}\sum_{i=1}^{n} i^2-\frac{1}{2}\sum_{i=1}^{n} i \\
&= \frac{n^3+n^2}{2}-\frac{n(n+1)(2n+1)}{12}-\frac{n(n+1)}{4} \\
&= \Omega(n^3)
\end{aligned}$$

after simplifications and elementary algebra.

(c) The given algorithm is clearly inefficient since during an iteration of the inner loop it often repeats several of the additions that have already been done in the previous iterations. Consider the following algorithm.

for i from 1 to n
    $B[i,i+1] \leftarrow A[i]+A[i+1]$
for k from 2 to $n-1$
    for i from 1 to $n-k$
        $j \leftarrow i+k$
        $B[i,j] \leftarrow B[i,j-1]+A[j]$

This algorithm works since the values $B[i,j-1]$ were already computed in the previous iteration of the outer for loop, when k was $j-1-i$, since $j-1-i<j-i$. The running time of the algorithm is $O(n)$ for the first for loop and $O(n^2)$ for the second set of loops, giving a total of $O(n^2)$ for the running time. Note that it is also $\Omega(n^2)$ since the algorithm computes at least $n(n-1)/2$ values.

5. (6 points) Properties of asymptotic notation

Let $f(n)$, $g(n)$, and $h(n)$ be asymptotically positive and monotonically increasing functions. For each of the following statements, decide whether you think it is true or false and give a proof or a counterexample.

(a) If $f(n)=\Omega(h(n))$ and $g(n)=O(h(n))$, then $f(n)=\Omega(g(n))$.

Answer: True. Since $f(n)=\Omega(h(n))$, there exist positive constants c_1 and n_1 such that $f(n)\geq c_1 h(n)$ for all $n\geq n_1$. Similarly, since $g(n)=O(h(n))$, there exist positive constants c_2 and n_2 such that $g(n)\leq c_2 h(n)$ for all $n\geq n_2$. This implies that for all $n\geq \max\{n_1,n_2\}$, $f(n)\geq c_1 h(n)\geq c_1 g(n)/c_2$. Therefore, $f(n)=\Omega(g(n))$.

(b) If $f(n)=O(g(n))$, then $f(n)^2$ is $O(g(n)^2)$.

Answer: True. Since $f(n)=O(g(n))$, there exist positive constants c and n_0 such that $f(n)\leq cg(n)$ for all $n\geq n_0$. We thus have $f(n)^2\leq c^2 g(n)^2$ for all $n\geq n_0$. Since c^2 is a positive constant, we thus have $f(n)^2=O(g(n)^2)$.