Solutions to Homework 1

22C:044 Algorithms, Fall 2000

1(a) Swapping two consecutive elements changes the relative order of those two elements only. The relative orders of all other pairs remains the same. Therefore, swapping two consecutive numbers that are in the wrong order removes exactly one inversion.

1(b) `Bubblesort` makes one swap and ten comparisons on input $1, 2, 4, 3, 5$.

1(c) `BetterBubbleSort` makes one swap and seven comparisons on input $1, 2, 4, 3, 5$.

1(d) The complexity of each iteration of the `while` -loop is $O(n)$. The while loop is executed at most $I + 1$ times, because the program exits the loop if no inversions are removed in a `while` -loop iteration. Therefore the complexity is $(I + 1)O(n) = O(nI)$.

1(e) False. The sequence $n, n-1, \ldots, 1$ is the only sequence with n numbers and $I = n(n-1)/2$ inversions. The running time on this input is $\Theta(n^2)$ but $nI = \Theta(n^3)$. So the running time is not $\Theta(nI)$.

1(f) The running time on input $n, 1, 2, 3, \ldots, n-1$ is $\Theta(n^2)$ but the input contains only $n - 1$ inversions, so that $n + I = 2n - 1 = \Theta(n)$.

2. One can use the following fact: if

$$a = \lim_{n\to\infty} \frac{f(n)}{g(n)}$$

exists (and if $f(n)$ and $g(n)$ are positive for all large n) then the relative growth rate of $f(n)$ and $g(n)$ can be deduced from the limit value a:

if $a = 0$ then $f(n)$ is $o(g(n))$, that is, $O(g(n))$ but not $\Theta(g(n))$,

if $a = \infty$ then $f(n)$ is $\omega(g(n))$, that is, $\Omega(g(n))$ but not $\Theta(g(n))$,

if $0 < a < \infty$ then $f(n)$ is $\Theta(g(n))$.

Using this observation, and the facts that $4^{\lg n} = n^2$, $\lg(n!) = \Theta(n \log n)$ and $\lg \sqrt{(n)} = \frac{1}{2} \lg n$ we obtain following classes:

$$
\begin{array}{rcl}
C_1 & = & \{1, 8 + 1/e^n\} \\
C_2 & = & \{\sqrt{\lg n}\} \\
C_3 & = & \{\lg \sqrt{n}\} \\
C_4 & = & \{n \log_{10} n, \lg(n!)\} \\
C_5 & = & \{n^2, 4^{\lg n}\} \\
C_6 & = & \{2^n\} \\
C_7 & = & \{3^n, 3^n + n^2\} \\
C_8 & = & \{n!\} \\
C_9 & = & \{(n+1)!\} \\
C_{10} & = & \{n^n, n^n + n!\}
\end{array}
$$

3(a) False. Choose $f(n) = 1, g(n) = n$. Then $f(n) + g(n) = n + 1$ is not $\Theta(\min(f(n), g(n))) = \Theta(1)$.

3(b) False. Choose $f(n) = 2^n$. Then $f(n/2) = 2^{n/2} = (\sqrt{2})^n$. We know that a^n is not $\Theta(b^n)$ if $a \neq b$.

3(c) False. Because

$$\lim_{n \to \infty} \frac{\sqrt{n} \lg n}{n} = 0,$$

$\sqrt{n} \lg n$ is $O(n)$ but not $\Theta(n)$.

3(d) False. Because

$$\lim_{n \to \infty} \frac{n^2 / \lg^2(n)}{n / \lg(n)} = \infty,$$

$n^2 / \lg^2(n)$ is $\Omega(n / \lg(n))$ but not $\Theta(n / \lg(n))$.

4(a) $A(n) = 1 + A(n-1) + A(n-2)$ for all $n \geq 3$.

4(b) (See the textbook on pages 36–37.) Notice first that $A(n) = F_n - 1$. This is certainly true for $n = 1$ and $n = 2$. And if it is true for $n - 1$ and $n - 2$ then it is also true for n because

$$A(n) = 1 + A(n-1) + A(n-2) = 1 + (F_{n-1} - 1) + (F_{n-2} - 1) = F_n - 1.$$

It follows from the principal of mathematical induction that for all positive integers n we have $A(n) = F_n - 1$.

The claim follows now from the formula

$$F_n = \frac{\phi^n - \hat{\phi}^n}{\sqrt{5}}$$

given in the text.

4(c) $A(n)$ is $\Theta(\phi^n)$.

4(d) `FastFibonacci(integer n)`

1. $a \longleftarrow 1$, $b \longleftarrow 1$
2. for $i \longleftarrow 3$ to n do
3. $\{\ tmp \longleftarrow a + b,\ a \longleftarrow b,\ b \longleftarrow tmp\ \}$
4. return b

The complexity of `FastFibonacci` is linear $\Theta(n)$.