COMP 3711 Design and Analysis of Algorithms
Fall 2014 Midterm Exam Solutions

1. 1.1 $(a) \to (b), (a) \to (c), (d) \to (a), (a) \to (e), (b) \leftrightarrow (c), (d) \to (b), (d) \to (c), (d) \to (e)$

   1.2 $\forall$ correct comparison-based sorting algorithm, $\exists\ c > 0$, $\exists\ n_0 > 0$, such that $\forall n > n_0$, $\exists$ an array of n elements for which the algorithm requires $\geq cn \log n$ comparisons to sort them.

   1.3 (a) $\Theta(n^{\log_3 5})$
   (b) $\Theta(n^2)$
   (c) $\Theta(3^n)$
   (d) $\Theta(n \log\log n)$

2. Let $X_i = 1$ if the i-the guess is correct, and 0 otherwise. Then the expected number of correct guesses is $E[X_1 + \cdots + X_n] = E[X_1] + \cdots + E[X_n]$.

   (a) In this case, $E[X_i] = 1/n$ for all i, so the answer is $n \cdot 1/n = 1$.
   (b) In this case, $E[X_i] = 1/(n-i+1)$, so the answer is $\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{2} + 1 = O(\log n)$.

3. Let $m = \lfloor n/2 \rfloor$. We look at $A[m]$ and $A[m+1]$. If $A[m] > A[m+1]$ or $A[m] > 0$, we recursively search the subarray $A[m+1..n]$. Otherwise we resursively search the subarray $A[1..m]$. The base case is when $n = 1$, then we return the only element in the subarray. The running time of the algorithm satisfies $T(n) = T(n/2) + 1$ and it solves to $T(n) = O(\log n)$.

4. 2 5 3 7 6 4 9 10 8

   3 5 4 7 6 8 9 10

   4 5 8 7 6 10 9

   5 6 8 7 9 10

   6 7 8 10 9

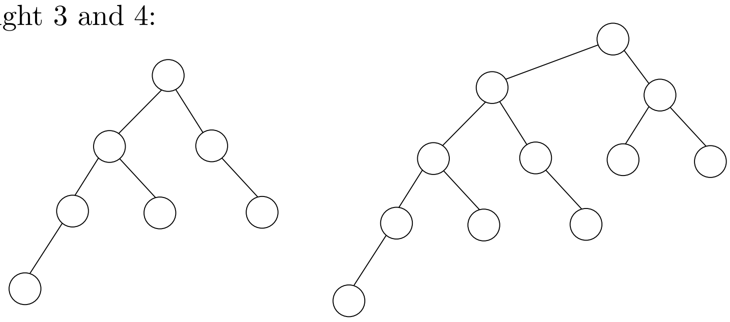
   7 9 8 10

   8 9 10

   9 10

   10

5. The trees of height 3 and 4:



$n_3 = 7$ $n_4 = 11$

Let n_h be the smallest size of a weight-balanced tree of height h. We have $n_h \geq n_{h-1} + \frac{n_{h-1}+1}{2} - 1 + 1 = \frac{3}{2}n_{h-1} + \frac{1}{2} > \frac{3}{2}n_{h-1}$. Solving the recurrence yields $n_h > (\frac{3}{2})^h$, so $h = O(\log n)$.