1. Do exercise 9.3-1, proving the bounds on the running time by substitution. This will explain why the deterministic linear-time selection algorithm used groups of size five.

2. Do problem 9.1. You can review priority queues in Chapter 6.5.

3. Do Problem 9.2, part e only. Assume all weights are one, so you can ignore parts a, b, c and d.

4. a) Show that the expected number of elements in the linked list at level \( i \) in a skip list, with \( n \) elements in the list at level zero, is \( np^i \) (where \( p \) is the probability, in the subroutine that chooses the random level for each item, that the level increases).

b) Consider the operation of merging two skip lists - taking two lists and combining them into a single skip list (which should store all the elements in sorted order). Describe how to merge two skip lists and use the result of part a) to bound the expected running time of your algorithm. You may assume all the elements in the skip lists are distinct.

5. Let \( X \) be a non-negative random variable, with \( \Pr[X > i] \leq (ni)^5(1/2)^i \), for all \( X \geq 1 \). Give the best asymptotic upper bound you can for \( E[X] \). If you like, you can assume that \( X \) is an integer random variable.

6. We are given a set \( R \) of \( n \) rectangles in the plane, with sides parallel to the \( x \) and \( y \) axes, represented by two intervals \( [x_{lo}, x_{hi}], [y_{lo}, y_{hi}] \). Describe an algorithm which outputs all pairs of rectangles which intersect. The running time of your algorithm will have to depend on the number \( k \) of intersecting pairs of rectangles, as well as \( n \). One good way to do this uses a skip list. Analyze the running time of your method.