50 marks	CS 301 Quiz 2	9:30-10:30 am, 30/10/7

Problem 1: Suppose you are given the sequence of keys 1,2,3,3,4,5,5 to sort. Consider an execution of randomized quicksort, (Remember that the quicksort algorithm divides the list into 3 lists: smaller than the pivot, equal to the pivot, and larger than the pivot.)

(a)[5 marks] What is the probability that the two keys having value 3 will get compared to each other?

The probability is 1.

(b)[5 marks] What is the probability that some 3 and some 5 will be compared to each other?

They will not get compared if 4 is chosen first among 3,3,4,5,5. Thus the probability of getting compared is 4/5.

Problem 2: Consider the problem of merging two increasing sequences $x = (x_1, x_2, x_3)$ and $y = (y_1, y_2)$. Suppose further that the values in x are distinct from those in y, and the algorithm is designed with this in mind for both parts below.

(a)[5 marks] How many leaves will a decision tree algorithm for this problem have? What does this say about the lower bound on the time taken by the decision tree algorithm? Precise numerical answers are expected. No partial credit. Be careful with log bases, floors and ceilings.

(b)[5 marks] Suppose x has length n, and y has length m. How many leaves does the decision tree algorithm have?

(b) m+n choose n. (a) 5 choose 2 = 10 leaves. 2 marks. So height is at least $\lceil \log_2 10 \rceil = 4$ = time taken. Log is to base 2 because we know that the values are distinct. 3 marks.

Problem 3: Suppose you are given as input integer arrays W[1..n], H[1..n], and integers x, y, p, where W[i],H[i] denote the width and height of the ith rectangle, x,y denote the width and height of a page, and p the number of pages. The goal is to determine whether the n rectangles can be fit into the given p pages such that no two rectangles overlap. Call this the AL problem – think of this as the problem of laying out advertisements in newspaper pages.

(a)[10 marks] Show that AL is in NP.

Certificate: page number and coordinates of corners etc. for each rectangle in the page it is going to be placed. Check in polytime that no rectangles overlap and are in a valid page, and do not overshoot the page boundaries.

(b)[10 marks] Show that AL is NP-hard. (Hint: suppose all rectangles have width exactly x, i.e. same as the width of the pages.)

reduction from bin packing, or partition. 7 marks for stating the instance map: rectangle height is item size, bin size is page height. 1 mark each to mention polytime, and each direction (c)[10 marks] Consider a special case of AL, called AL1, in which p=1, i.e. there is only one page available. Prove that AL1 is NP-hard.

reduce from partition. rectangle height = item size. page height = half the sum of heights, rectangle width = some very small number, page width = 2 times that very small number.