CS 5633: Analysis of Algorithms

Homework 3

1. For this problem, use of the proper notation is worth half the points. Clearly define your random variables.

   (a) What is the expected value of a single roll of a fair, 6-face die.

   (b) Use indicator random variables to compute the expected value of the sum of n dice.

2. Suppose you have the option of playing a game in which two fair, 6-face dice are rolled. If you roll two 1's, you win \$10. If you roll one 1, you win \$1. If you do not roll a 1, you pay 50 cents. What is the expected value of the game from your perspective?

3. When Randomized Quicksort runs, how many calls are made to the random number generator in the

   (a) Best Case?

   (b) Worst Case?

   (c) Average Case?

4. Consider the quicksort algorithm on an input which may contain the same value multiple times.

   (a) How does deterministic quicksort behave on an array with n equal keys? What is its runtime? What is the behavior and the runtime of randomized quicksort in this case? Justify your answer.

   (b) If you change $A[j] \leq x$ to $A[j] < x$ in the pseudocode for partition, how does quicksort behave on an array with n equal keys? What is its runtime?

   (c) How does deterministic quicksort behave on an array with just two distinct keys (the total number of keys is still n)?

   (d) Give pseudo-code for a 3-way partition that partitions the array into three parts: keys less than the pivot, keys equal to the pivot, keys greater than the pivot. Your code should be in-place (so it should use at most constant extra storage) and it should run in linear time.

   (e) Consider an implementation of quicksort which uses 3-way partition and only recurses on the portions of the array with keys less than the pivot and with keys greater than the pivot. If the array of n keys contains only 2 different values, what is the worst-case runtime?

(f) If the array contains n keys of d different values, show that the worst-case runtime of this variant of quicksort that uses 3-way partition is $O(dn)$.