Homework #5: Randomized Algorithms
Due Date: Tuesday, March 31, 1998 (5:00pm)

Revised Late Policy

For the first half of the course, homeworks were due at 9:00 a.m. on the due date, and late homeworks were accepted up until 5:00 p.m. that same day, with a penalty.

For the second half of the course, we will accept homeworks up until 5:00 p.m. *with no penalty* (i.e. they are now due at 5:00 p.m.). However, we will not accept homeworks any later than that time, except with previous arrangements.

Collaboration Policy

You may collaborate with others on this entire homework. However, please re-read the policy guidelines from Handout #1.

Reading

Review Ch. 6.2, 6.3, and 6.4 of CLR for a review on probability, if necessary. Read Ch. 8.3, 8.4, 10.2, and 13.4 of CLR.

Practice

Recall, these exercises are purely for your own practice. You need only turn in the official problems.

- Do CLR 6.2-3
- Do CLR 6.2-4
- Do CLR 6.2-10, 6.2-11. (These are *great* problems!)
- Do CLR 8.2-5.
- Do CLR 11.3-4

Problems

1. Variant of CLR 6-2 (25 points).
   The following program determines the maximum value in an unordered array $A[1..n]$.

   `Find-Max`(A, n)
   1 `Randomly-Permute`(A, n)
   2 $max \leftarrow -\infty$
   3 **for** $i \leftarrow 1$ **to** n
   4   **do** ▷ Compare $A[i]$ to max.
   5   **if** $A[i] > max$
   6   **then** $max \leftarrow A[i]$

   The call in line 1 re-orders the elements in the array by choosing a uniformly random permutation of the items (see Extra Credit problem). We want to determine the expected number of times the assignment in line 6 is executed. Assume that all array entries are distinct.

   a. If a number x is randomly chosen from a set of i distinct numbers, what is the probability that x is the largest number in the set?
   b. When line 6 of the program is exectued, what is the relationship between $A[i]$ and $A[j]$ for $1 \leq j \leq i$.
   c. For each i in the range $1 \leq i \leq n$, what is the probability that line 6 is exectued? Justify your answer.
   d. Let $s_1, s_2, \ldots, s_n$ be n random variables, where s_i represents the number of times (0 or 1) that line 6 is exectued during the ith iteration of the **for** loop. What is $E[s_i]$?
   e. Let $s = s_1 + s_2 + \cdots + s_n$ be the total number of times that line 6 is exectued during some run of the program. Show that $E[s] = \Theta(\lg n)$.

2. (30 points)
   In this problem, we give yet another proof that randomized `Quicksort` runs in $O(n \lg n)$ expected time. The text gives a proof of this fact in Chapter 8.4, by giving a recurrence relation which involves a summation over the possible partitions. In lecture, we say a simpler argument based on labelling partitions as "good" or "bad". In this problem, we will get a tighter upper bound on the expected number of comparisons which are performed, by looking at the chance that any given two elements are directly compared to each other.

Input: $A = \{a_1, \cdots, a_n\}$.

RQUICKSORT(A).
  if $|A| = 1$, then **return** A.
  else
    Choose *pivot* element uniformly at random from A.
    Compare *pivot* to all other elements. Let
      $A_1 = \{\text{elements smaller than } pivot\}$
      $A_2 = \{\text{elements larger than } pivot\}$.
    return $\{\text{RQUICKSORT}(A_1), pivot, \text{RQUICKSORT}(A_2)\}$.
end.

Assume that the sorted order of the elements is given by $\Pi = \{\pi(1), \pi(2), \ldots, \pi(n)\}$. For each possible pair of unique elements i and j with $1 \leq i < j \leq n$, we define a 0-1 random variable x_{ij} as follows:

$$x_{ij} = \begin{cases} 1 & \text{if } \pi(i) \text{ and } \pi(j) \text{ are directly compared to each other} \\ 0 & \text{otherwise} \end{cases}$$

a. Prove that $E[x_{ij}] = \frac{2}{j-i+1}$. (*Hint:* To get started, think about some special cases, such as the chance that $\pi(1)$ and $\pi(n)$ are compared, or $\pi(5)$ and $\pi(6)$, or $\pi(8)$ and $\pi(10)$.

b. Let C be a random variable equal to the overall number of comparisons performed during a single run of `Randomized-Quicksort`. Give a simple formula for C in terms of the various x_{ij} variables. Then use you formula, and the linearity of expectations, to get an expression for $E[C]$, using summations.

c. Use your above expression to prove that $E[C] \leq 2nH_n$, where H_n is the Harmonic Number, defined on page 44.

3. Do CLR 11-3. (45 points)
You do not need to do part (d). You should use the results as a given for other parts. Before starting this problem, make sure you understand the *multiple-array representation of objects* on p. 209.

Extra Credit

Page 161 described a function `Random`(a, b), which returns an integer between a and b inclusive, with each such integer being equally likely. If programming in C, this would be similar to getting the value $a + (random()\%(b - a + 1))$.

a. Assume that `Random` is a constant-time function for reasonable values of a and b. Give a $\Theta(n)$-time randomized algorithm, which creates a uniformly random permutation of an array $A[1..n]$ of items.

b. Prove the correctness of your algorithm. Do not argue vaguely that all outcomes are equally likely; instead, prove formally that the probability of getting a particular permutation Π, is exactly $1/n!$.

c. Assume instead that your only source of randomness is a constant-time function `Random-bit`, which returns either 0 or 1 with equal probability. Argue that any algorithm which randomly permutes the array requires $\Omega(n \lg n)$ time.