CPSC-320 Intermediate Algorithm Design and Analysis Summer 2003

Homework # 2

Due Monday, May 26 at 6pm.

NAME:____

Signature:____

STD. NUM: ____

General guidelines for homeworks:

Before starting on this homework, review the homework guidelines provided on the first day of class (also on the web under "Course Description"). Remember that it is encouraged to discuss the problems with others in the class, but all write-ups are to be done on your own. **Homework grades will be based not only on getting the "correct answer," but also on good writing style and clear presentation of your solution.** It is your responsibility to make sure that the graders can easily follow your line of reasoning.
Try every problem. Even if you can't solve the problem, you will receive partial credit for explaining why you got stuck on a promising line of attack. More importantly, you will get valuable feedback that will help you learn the material.
Please acknowledge the people with whom you discussed the problems and what sources you used to help you solve the problem (e.g. books from the library). This won't affect your grade but is important as academic honesty.

1. We are given a list, L, of n elements and one more element, X, and the problem is to decide whether X is in L. If X is in L we have to identify at least one element of L that it's equal to. One algorithm that solves this problem is linear search:

```
LINEAR_SEARCH (L,lower,upper,X)
   {Look for X in L[lower..upper]
    Report its position if found, else report 0.}

   if L[lower] = X then
      return lower
   else
      if lower = upper then
         return 0
      else
         return LINEAR_SEARCH(l,lower+1,upper,X)
      end
   end
```

(a) Derive a recurrence relation for the cost of this algorithm in terms of the number of comparisons. Then solve this recurrence to obtain an analytical expression for the worst cost of this algorithm.

(b) Assume that X is equally likely to be any element of L, if it's in L. Let p_i be the probability that X equals $L[i]$, $i = 1, \ldots, n$, and let p_0 be the probability that X is not in L. Derive an expression for the average cost in terms of n and p_0. What are sensible upper and lower bounds on this cost?

2. Let's assume that the list L has been sorted so that $L[1] < X < L[n]$. Then X should be close to $L[\lceil pn \rceil]$, where

$$p = \frac{X - L[1]}{L[n] - L[1]}$$

- Why is the above statement true? Try a simple example.

- Suppose we have an algorithm that first probes $L[\lceil pn \rceil]$. If $X < L[\lceil pn \rceil]$, then it sequentially probes the elements $L[\lceil pn - i\sqrt{n} \rceil]$, $i = 1, 2, 3, \ldots$, until it finds the smallest i for which $X \geq L[\lceil pn - i\sqrt{n} \rceil]$. Similarly, if $X > L[\lceil pn \rceil]$, then it sequentially probes the elements $L[\lceil pn + i\sqrt{n} \rceil]$, $i = 1, 2, 3, \ldots$, until it finds the smallest i for which $X \leq L[\lceil pn + i\sqrt{n} \rceil]$. When this jump search ends, we know X's position to within roughly $\sqrt{n}$ elements. That is, we have reduced the original problem to one whose size is the square root of the size of the original one. Finally, we call our algorith recursively with the sublist. What is the computational cost of this algorithm?

3. The following program determines the maximum in an unordered array $A[1..n]$

```
1  max = -infinity {or a very small number}
2  for i = 1 to n
3     compare A[i] to max.
4     if (A[i] > max) then
5        max = A[i]
6     end
7  end
```

Our goal is to determine the expected number of times that line 5 is executed. We assume that the elements of A are unique and drawn uniformly at random.

(a) If a number is randomly chosen from A, what is the probability that this element is the maximum element?

(b) What is the relation between $A[i]$ and $A[j]$, $j = 1, 2, \ldots, i$, when line 5 is executed.

(c) For i in the range $1 \le i \le n$, what is the probability that line 5 is executed?

(d) Let $s_1, s_2, \ldots, s_n$ be n random variables, where $s_i = 1$ if line 5 is executed and $s_i = 0$ otherwise. What is $\mathbb{E}(s_i)$?

(e) Let $s = s_1 + s_2 + \ldots + s_n$ be the total number of times that line 5 is executed. Prove that $\mathbb{E}(s) = \Theta(\lg n)$.