CS3432.03 (W99)

Problem Set # 1

1. Themes: Divide and Conquer; Recurrence Relations; Generating Functions

1. Modify *binary search* of a sorted array $A[1\dots n]$ so that the "middle" item probed first is at location $\lfloor n/2 \rfloor$ rather than $\lfloor (n+1)/2 \rfloor$.

(1) Derive the recurrence that gives the worst case run-time (in terms of number of comparisons), $T(n)$.

(2) Solve the recurrence *exactly* (*not* in Big-O notation), providing full details throughout.

Note. You will encounter a need to assume that

$$\left\lceil \frac{\left\lceil \frac{a}{b} \right\rceil}{c} \right\rceil = \left\lceil \frac{a}{bc} \right\rceil$$

for all integers $a, b, c \geq 1$. Please don't "assume"; prove this.

2. Given the generating function $G(z) = a_0 + a_1 z + a_2 z^2 + \dots$.

Find (in terms of $G(z)$, in closed form) the generating function of the sequence $a_0, a_0 + a_1, a_0 + a_1 + a_2, a_0 + a_1 + a_2 + a_3, \dots$.

Use **two** methods, one of which uses the *convolution*.

Hint. The answer is

$$\frac{G(z)}{1-z}$$

but still you are required to discover two different proofs.

2. Theme: Miscellaneous

1. Consider the program-segment below:

$r \leftarrow 0$
for $i = 1$ **to** n **do**
  begin
   $r \leftarrow r + 1$
  end

Find the tightest possible upper bound for the number of *bit operations* (assume that numbers are held in binary internally) that are *due to the repeated execution of* $r \leftarrow r + 1$ *only*—that is, ignore the bit operations that are required to maintain the loop.

Hint. The "pessimistic" answer is $O(n \log n)$, but you must be able to do much better! At some point you might find it helpful to prove and use the fact that $\sum_{i \geq 0} i/2^i = O(1)$. And, as always, you may start by ignoring the hint.