CS 6363: Homework 1

Version 1.2

Instructor: Benjamin Raichel

Due Sunday September 11, at 11:59 pm

Homework is to be submitted online through eLearning. Typeset solutions (in LaTeX) are strongly preferred, but not required. If you chose to hand wright and scan your solutions, if what is written is illegible you will be marked down accordingly. Please make sure your name is on each page of the submission. You do not need to restate the problem, just the problem number.

Explanations are to be given for each problem, unless the problem specifically states it is not necessary. Explanations should be clear and concise. Rambling and/or imprecise explanations will be marked down accordingly.

Algorithm Analysis and Induction

(Total of 20 points)

Problem 1. Running Time Analysis

Provide tight asymptotic bounds (i.e. $\Theta(\cdot)$) on the worst case running times of the following two procedures. You do not need to explain your bounds. (3 points each)

(a)

```
procedure BackForwardAlg(n)
    if n ≤ 10 then
        return n
    if n even then
        return BackForwardAlg(n/2)
    else
        return BackForwardAlg(n + 3)
```

(b)

```
procedure RecursiveAlg(A[1 . . . n])
    if n == 1 then
        return False
    mid = ⌈n/2⌉
    for i = 1 to mid do
        for j = mid + 1 to n do
            if A[i] == A[j] then
                return True
    return (RecursiveAlg(A[1 . . . mid]) || RecursiveAlg(A[mid + 1 . . . n]))
```

(c) (2 bonus points) Give a one sentence description of what *RecursiveAlg* does.

Problem 2. Induction on Binary Trees

(a) (9 points) In the following a node is a data type with two associated pointers labeled "left" and "right". We call a node a leaf if both the left and right pointers are null. A "good binary tree" has the following recursive definition.

$$gbt = \begin{cases} \text{A leaf node.} \\ \qquad \textit{or} \\ \text{A node, } r\text{, such that } r.\text{left and } r.\text{right point to distinct } gbt\text{'s.} \end{cases}$$

Prove, using induction, that a gbt with $n > 0$ leaves, has $2n - 1$ nodes in total (i.e. internal and leaf nodes together). Note that every gbt has a root, i.e. a single node with no incoming pointers.

(b) (5 points) A "bad binary tree" is defined to also allow for the empty tree. Specifically,

$$bbt = \begin{cases} \text{Null (i.e. the empty tree)} \\ \qquad \textit{or} \\ \text{A leaf node.} \\ \qquad \textit{or} \\ \text{A node, } r\text{, such that } r.\text{left and } r.\text{right point to distinct } bbt\text{'s.} \end{cases}$$

Does it hold that a bbt with $n > 0$ leaves, has $2n - 1$ nodes in total? If yes, prove it. If no, what is the best upper bound possible on the total number of nodes in terms of n.

Functions and Asymptotic Bounds

(Total of 40 points)

Problem 3. Asymptotic Bounds

Provide tight asymptotic bounds (i.e. $\Theta(\cdot)$) on the following. You do not need to explain your bounds. (3 points each)

(a) $\sum_{i=1}^{n} n/i$

(b) $\sum_{i=1}^{n} i^3$

(c) $\sum_{i=1}^{n} \log(n/i)$ [Hint: Stirling's approximation may be useful (see Wikipedia).]

(d) $2/n + \sqrt{n} + \log^3 n$

(e) The number of bits needed to write 10^n in binary.

(f) Planet X has a current population 1 billion. Assuming that the population of planet X doubles every year, give an asymptotic bound on the population of planet X in 100 years from now.

Problem 4. Ordering functions

(22 points) Sort the following 25 functions from asymptotically smallest to asymptotically largest, indicating ties if there are any. You do not need to turn in proofs (in fact, please *don't* turn in proofs). Some are difficult, don't panic if you cannot get them all.

$n^{4.5} - (n-1)^{4.5}$	n	$n^{2.1}$	$\lg^*(n/8)$	$1 + \lg\lg\lg n$
$\cos n + 2$	$\sqrt{\lg n}$	$\lg n$	$(\lg^* n)^{\lg n}$	n^5
$\lg^* 2^{2^{2^n}}$	$2^{\lg n}$	$\sqrt{n}^e$	$\sum_{i=1}^n i$	$\sum_{i=1}^n i^2$
$n^{\frac{7}{(2n)}}$	$n^{3/(2\lg n)}$	$12 + \lfloor \lg\lg(n) \rfloor$	$(\lg(2+n))^{\lg n}$	$\left(1 + \frac{1}{154}\right)^{15n}$
$n^{1/\lg\lg n}$	$n^{\lg\lg n}$	$\lg^{201} n$	$n^{1/125}$	$n \lg^4 n$

To simplify notation put all the functions in a single column, where $f(n)$ being above $g(n)$ in the column means $f(n) = o(g(n))$. If $f(n) = \Theta(g(n))$, then write $f(n)$ and $g(n)$ separated by a comma on the same line. For example, the functions n^2, n, $\binom{n}{2}$, n^3 should be written

- n
- n^2, $\binom{n}{2}$
- n^3

Hint: When considering two functions $f(n)$ and $g(n)$ it is sometimes useful to consider the functions $\log f(n)$ and $\log g(n)$, since $\log f(n) = o(\log g(n))$ implies $f(n) = o(g(n))$ (but be careful, the implication does not work in the other direction, and in particular $\log f(n) = O(\log g(n))$ does NOT imply $f(n) = O(g(n))$). Also the relations $(1 + x) \leq e^x$ for all x, and $(1 + x) \geq e^{x/2}$ for $0 \leq x \leq 1/2$, may be useful.

Recurences

(Total of 40 points)

Problem 5. UTD Parking

(10 points) To help with parking issues, UTD has just created a new parking lot consisting of n spaces in a single long row. Rather than creating separate spots for motorcycles and cars, management decided to make all n spaces small, such that motorcycles will occupy one space and cars will occupy two. Write a recurrence $P(n)$, counting the number of possible ways to park cars and motorcycles in this parking lot such that the lot is full. Specifically, for a full lot we can write a string mccmmmcmc.... representing the left to right ordering of appearance of 'm'otorcycles and 'c'ars, and we wish to count the number of distinct strings which represent a full lot. For simplicity, you can assume $P(0) = 1$.
Hint: The recurrence should have a simple form. Also, don't forget to include base cases.

Problem 6. Bounding Recurrences

Provide tight asymptotic upper and lower bounds on the following recurrences (i.e. $\Theta(\cdot)$). If you cannot give matching upper and lower bounds, give the best upper and lower bounds that you know how to. Correct answers without explanation are given full credit, but I suggest you give a brief justification for each problem, both for practice and for possible partial credit.

(a) (5 points) $T(n) = 2T(n/2) + n^4$

(b) (5 points) $T(n) = 16T(n/4) + n^2$

(c) (5 points) $T(n) = 2T(n/3) + T(n/4) + n$

(d) (5 points) $T(n) = T(n-1) + \sqrt{n}$

(e) (5 points) $T(n) = 3T(n/2) + 5n$

(f) (5 points) $T(n) = T(\sqrt{n}) + 7$