CS 6363: Homework 1

Version 1.0

Instructor: Benjamin Raichel

Due Sunday September 9, at 11:59 pm

Homework is to be submitted online through eLearning by the above deadline. 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. 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.

Problem 1. Running Time Analysis (8 points)

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. (4 points each)

(a)

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

(b)

```
procedure RecursiveAlg($A[1 \dots n]$)
    if $n == 1$ then
        return False
    $mid = \lceil n/2 \rceil$
    for $i = 1$ to $mid$ do
        for $j = mid + 1$ to $n$ do
            if $A[i] == A[j]$ then
                return True
    return (RecursiveAlg($A[1 \dots mid]$) || RecursiveAlg($A[mid+1 \dots n]$))
```

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

Problem 2. Asymptotic Bounds (12 points)

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

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

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

(c) 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.

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

Problem 3. Ordering functions (20 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).

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

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 $f(n) = O(g(n))$).

Problem 4. 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. You should assume $P(0) = 1$.
Hint: The recurrence should have a simple form. Also, don't forget to include base cases.

Problem 5. Bounding Recurrences (20 points)

Provide tight asymptotic upper and lower bounds for four of the following six recurrences (i.e. $\Theta(\cdot)$). Clearly label which four you have chosen to solve.

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) $T(n) = 2T(n/2) + n^4$

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

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

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

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

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

Problem 6. Peaked Arrays (10 points)

Let $A[1 \dots n]$ be an array of n distinct integers. We say that A is *peaked*, if there exists an index $1 \leq k \leq n$ such that:

(1) For all $1 \leq i < k$, $A[i] < A[i+1]$ and (2) for all $k < i \leq n$, $A[i-1] > A[i]$.

That is, the array increases until k and decreases afterward. Call the index k the *peak* of A.

Describe an algorithm which in $O(\log n)$ time finds the peak of a given peaked array $A[1 \dots n]$ consisting of n distinct integers. Specifically, give short descriptions of how your algorithm works, why it is correct, and why it achieves the stated running time.

Problem 7. Inversions (20 points)

An *inversion* in an array $A[1 \dots n]$ is a pair of indices $\{i, j\}$ such that $i < j$ and $A[i] > A[j]$. The number of inversions in an n-element array is between 0 (if the array is sorted) and $\binom{n}{2}$ (if the array is sorted backward). Describe and analyze an algorithm to count the number of inversions in an n-element array in $O(n \log n)$ time. *[Hint: Modify mergesort.]*