Homework #3

You do not need to turn in these problems. The goal is to be ready for the in-class quiz that will cover the same or similar problems, and to prepare you for the exams.

Problem 1: Algorithm Analysis

Consider the following basic problem. You're given an array A consisting of n integers $A[1], A[2], \ldots, A[n]$. You'd like to output a two-dimensional n-by-n array B in which $B[i, j]$ (for $i < j$) contains the sum of array entries $A[i]$ through $A[j]$—that is, the sum $A[i]+A[i+1]+\cdots+A[j]$. (The value of array entry $B[i, j]$ is left unspecified whenever $i \geq j$, so it doesn't matter what is output for these values.) Below is a simple algorithm to solve this problem:

```
1  for i = 1 to n
2        for j = i + 1 to n
3        Add entries A[i] through A[j]
4        Store the result in B[i, j]
```

(a) Give a function $f(n)$ that is an asymptotically tight bound on the running time of the algorithm above. Using the pseudocode above, argue that the algorithm is, in fact $\Theta(f(n))$.

(b) Although the algorithm you analyzed above is the most natural way to solve the problem, it contains some highly unnecessary sources of inefficiency. Give a different algorithm to solve this problem with an asymptotically better running time than the provided algorithm.

(c) What is the running time of your new algorithm?

Problem 2: Try sorting

Given a set A of n distinct positive integers and another integer t, describe an algorithm that determines whether or not there exists two elements in A such that their product is exactly t. Come up with an $O(n \log n)$ algorithm to solve this problem. Hint: When you see a runtime target that looks familiar, try to think of other algorithms you can use as subroutines in your algorithm with the same time complexity to make your life easier. In this case, try sorting the list first.

Problem 3: Algorithms and decision trees

You are given 9 identical looking balls and told that one of them is slightly heavier than the others. Your task is to identify the defective ball. All you have is a balanced scale that can tell you which of two sets of balls is heavier.

(a) Show how to identify the heavier ball in just 2 weighings.

(b) Give a decision tree lower bound showing that it is not possible to determine the defective ball in fewer than 2 weighings.

Problem 4: Heap algorithms

Given k sorted lists, merge them into one sorted list in $\mathcal{O}(n \log k)$ time where n is the sum of the lengths of all lists (i.e., the total number of elements in the input).

Problem 5: Median-heap data structure

Design a"Median-heap" data structure, which supports extract-median operation in $\mathcal{O}(1)$ time and insert/delete operations in $\mathcal{O}(\log n)$ time. Recall the definition of median (intuitively, the median of a list is the middle element when that list is sorted). If the list has even length, you can choose the larger of the two middle elements as the median.