Fundamental Algorithms
Courant, Fall 2017

Homework 1

Instructor: Amir Shpilka
Due Date: Wed Sep 13, 5:00pm

1. *(Induction)*
   (a) (3 points) Prove that $1^3 + 2^3 + \cdots + n^3 = (1 + 2 + \cdots + n)^2$.
   (b) (3 points) Prove, using induction, that the product of arbitrarily many odd numbers $x_1, x_2, \ldots, x_n$ is itself odd.
   (c) (4 points) Assume you have a rectangular chocolate bar divided into a $m \times n$ grid. You want to break it into mn pieces to distribute to the hungry undergrads in the classroom next door, but you are a lazy grad student and want to minimize the number of times you break a piece of chocolate along a grid line. Prove that the minimum number of breaks you need to end up with 1×1 pieces is $mn - 1$.
   (d) (5 points) (**Bonus**[1]) You now have a $2^n \times 2^n$ chocolate bar that you need to distribute among the faculty members. The head of the department, in a quintessentially eccentric way, wants to eat only one particular (arbitrary) square. The rest of the faculty don't care which squares they get, except that they each want 3 squares in an L-shape. By some coincidence, the number of squares is exactly enough for the number of faculty members. Prove that there is some way to break up the chocolate bar, for any choice of n and the department head's square.

2. You are given an unordered sequence of n numbers in an array, and you want to find the median value of the sequence.
   (a) (3 points) Describe, in English, how you would find the median of the numbers.
   (b) (3 points) Write pseudocode to implement your algorithm.
   (c) (4 points) If each time your algorithm reads an array element it costs \$1, then calculate the cost of finding the median in the *worst case*, in terms of n.

3. You and your n best friends decide to meet at the beach on Saturday. However, since they're all immensely popular, they each tell you they can only be at the beach for a certain time period. You, not wanting to miss out on hanging out with any of them, plan to be at the beach whenever any of them are there. But since you actually don't like being at the beach, you don't want to be there for any more time than necessary.

   Assuming you wrote some code to parse all their emails and have populated an array `a[]` with each friend's arrival time, and an array `b[]` with their departure times, how would you calculate your optimal arrival and departure time?

   I should be at the beach when at least one of the friends is at the beach and i can only go and come from the beach once (in total). Don't use min or max functions.

   (a) (3 points) Write an informal description of your algorithm.
   (b) (3 points) Give a pseudocode implementation of your algorithm.
   (c) (4 points) If each time your algorithm compares two numbers it costs \$1, then calculate the cost of your algorithm in terms of n.

4. Let $A[1, \ldots, n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called a *transposition* of A.
   (a) (2 points) List all transpositions of the array $\langle 7, 4, 1, 6, 8 \rangle$.
   (b) (3 points) Which arrays with distinct elements from the set $\{1, 2, \ldots, n\}$ have the smallest and the largest number of transpositions and why? State the expressions *exactly* in terms of n.

[1] Bonus questions are not mandatory, and not attempting them will not harm you in any way. They are often more difficult (so that you don't complain the homework isn't challenging enough) and/or teach extra material.

(c) (5 points) What is the relationship between the running time of INSERTION-SORT and the number of transpositions I in the input array? *Justify your answer*.

(d) (5 points) (**Bonus**) Let $A[1, \ldots, n]$ be a random permutation of $\{1, 2, \ldots, n\}$. What is the expected number of transpositions of A. What can you conclude about the average case running time of INSERTION-SORT (where the average is over all arrays A of size n)?

Hint: Recall the linearity of expectation, i.e., for any real a, b, c and any random variables X, Y,

$$E(aX + bY + c) = aE(X) + bE(Y) + c\ .$$