

Homework 2

Due February 4

Turn in your assignment to Canvas by the start of class time on the due date. Late assignments are not accepted.

1. Here is a $\Theta(n)$ algorithm that, given a sorted array, A , of n integers and another integer x , determines whether or not there exists two distinct elements in A whose sum is exactly x . Note that I use $0 \dots n - 1$ for my array indices.

```
high = n-1
low = 0
found = false
while ((low < high) AND NOT found) {
    if (A[low] + A[high] == x) then
        found = true
    else if (A[low] + A[high] < x) then
        low++
    else
        high--
}
```

I want you to prove the correctness of the above algorithm. You should proceed in two cases. Since the while loop condition contains an AND one of the two halves of that condition must be false to cause termination. Thus when the loop ends, either:

- a) **found** is true. In this case I want you to conclude that **low** and **high** are the indices of two distinct elements that sum to **x**. This should be pretty obvious but I still want you to give a short justification of this claim.
- b) **low** \geq **high**. In this case I want you to conclude that there does not exist two distinct elements in A that sum to x . This is not so obvious. To prove this claim use the following loop invariant:

Let S be the set of elements in A in the ranges $[0 \dots low - 1]$ and $[high + 1 \dots n - 1]$.
At the start of each iteration of the while loop, no element in S is part of a pairing of distinct elements in A that sums to x .

2. Problem 2-4 “Inversions” on page 41-42 of your text. For part (d) you have to give a clear argument as to why your algorithm is correct (you do not need to use a loop invariant but you can) and you also have to justify the running time of your algorithm.

You might be asking yourself: “why do we care so much about inversions?” We will talk about this in class next week but to whet your appetite it can be used to measure similarity of people in a collaborative filtering system like Netflix or Amazon recommendation systems.

3. For this problem you are working with a deck of n cards numbered $1 \dots n$ that has been shuffled to produce a random permutation of the cards. Furthermore, to make things interesting for part (c) assume that the even numbered cards are printed in black ink while the odd numbers are printed in red ink.
- (a) Suppose your friend takes the randomly shuffled deck and turns over one card at a time. Before each card is turned over you predict what it is. Since you don't actually have psychic powers, and you're not good at remembering which cards were turned over already, you simply guess a number at random from 1 to n each time (**note: this means that you may guess 34 even if 34 was turned over already**). What is the expected number of predictions that you get correct? Prove your answer is correct using indicator random variables.
 - (b) Now suppose your psychic abilities are still undeveloped but you have perfected the practice of remembering exactly which cards have been turned over already. Using this new skill, each time you make a guess it will be a random choice from the set of cards that have not been turned over yet. What is the expected number of predictions that you get correct in this scenario? Prove your answer is correct using indicator random variables.
 - (c) Try as you might, your psychic powers are still undeveloped so you've decided to go to the dark side and resort to cheating. Before your friend comes over you put a dot of "invisible ink" on the back side of each black card (and remember that even numbered cards are printed in black). Suppose you can see this dot if you put on your polarized sunglasses. As in part (b), we'll assume that your memory skills are perfect. Thus, you can remember every card that has been turned over and you can tell whether the upcoming card is an even or odd number. What is the expected number of predictions that you get correct in this scenario?
 - (d) Now suppose we lay out the n cards on a table from left to right. Let's define an "inversion" on two cards the same way we defined inversions on array elements in Problem 2 above. Namely, if card i is to the right of card j but the number on card i is less than the number on card j we will say card i and card j form an inversion. Assuming we have a randomly shuffled deck what is the expected number of inversions? Prove your answer is correct using indicator random variables.