Problem Set 2

Due dates: Typeset your solution in LaTeX. Electronic submission of the resulting .pdf file of this homework is due on **Friday, Feb 3, before 11:59pm** on canvas. If your submission cannot be checked by turnitin, then it will not be graded.

Name: (put your name here)

Resources. (All people, books, articles, web pages, etc. that have been consulted when producing your answers to this homework)

On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment. Furthermore, I have disclosed all resources (people, books, web sites, etc.) that have been used to prepare this homework.

Signature: ____________________________

Problem A. Solve the following five subproblems.

Problem 1 (20 points)**.** Give a self-contained proof of the fact that

$$\log_2(n!) \in \Theta(n \log n).$$

[For part of your argument, you can use results that were given in the lecture, but you should write up the proof in your own words. Make sure that you write it in complete sentences, even when the sentence contains formulas. A good check is to read out the entire solution aloud. It should read smoothly.]

Solution.

Problem 2 (20 points)**.** Amelia attempted to solve n algorithmic problems. She wrote down one problem per page in her journal and marked the page with 😐 when she was unable to solve the problem and with 🙂 when she was able to solve it. So the pages of her journal look like this:

😐	😐	😐	🙂	😐	🙂	😐	🙂

Use the decision tree method to show that any algorithm to find a page with an 🙂 smiley on has to look at all n pages in the worst case.

Solution.

Problem 3 (20 points)**.** Amelia attempted to solve n algorithmic problems. She wrote down one problem per page in her journal and marked the page with 😐 when she was unable to solve the problem and with 🙂 when she was able to solve it. So the pages of her journal look like this:

😐	😐	😐	🙂	😐	🙂	😐	🙂

Use an adversary method to show that any method to find a page with a 🙂 smiley on it might have to look at all n pages.

Solution.

Problem 4 (20 points)**.** Amelia attempted to solve n algorithmic problems, where n is an odd number. She wrote down one problem per page in her journal and marked the page with 😐 when she was unable to solve the problem and with 🙂 when she was able to solve it. Suppose that we want to find the pattern 😐🙂, where she was unable to solve a problem, but was able to solve the subsequent problem.

Find an algorithm that always looks at fewer than n pages but is able to correctly find the pattern when it exists. [Hint: First look at all even pages.]

Solution.

Problem 5 (20 points)**.** Suppose that we are given a sorted array $A[1..n]$ of n numbers. Our goal is to determine whether or not the array A contains duplicate elements. We will limit ourselves to algorithms that use only the spaceship operator `<=>` for comparisons, where

```
a <=> b :=
  if a < b then return -1
  if a = b then return  0
  if a > b then return  1
  if a and b are not comparable then return nil
```

No other methods will be used to compare or inspect elements of the array.

(a) Give an efficient (optimal) comparison-based algorithm that decides whether $A[1..n]$ contains duplicates using the spaceship operator for comparisons.

(b) Use an adversarial argument to show that no algorithm can solve the problem with fewer calls to the comparison operator `<=>` than the algorithm that you gave in (a).

Solution.

Checklist:
- ☐ Did you add your name?
- ☐ Did you disclose all resources that you have used?
  (This includes all people, books, websites, etc. that you have consulted)
- ☐ Did you sign that you followed the Aggie honor code?
- ☐ Did you solve all problems?
- ☐ Did you write the solution in your own words?
- ☐ Did you submit the pdf file of your homework?