

CS 3000: Algorithms & Data — Spring 2024

Sample Solutions to Homework 1

Due Monday January 22 at 11:59pm via Gradescope

Name:

Collaborators:

- Make sure to put your name on the first page. If you are using the \LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This homework is due Monday January 22 at 11:59pm via Gradescope. No late assignments will be accepted. Make sure to submit something before the deadline.
- Solutions must be typed. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. We recommend that you use \LaTeX , in which case it would be best to use the source file for this assignment to get started. Your submitted file must be a PDF file.
- We encourage you to work with your classmates on the homework problems, but also urge you to attempt all of the problems by yourself first. If you do collaborate, you must write all solutions by yourself, in your own words. Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly forbidden.

Problem 1. ($4 + 8 = 12$ points) What does this code do?

You encounter the following mysterious piece of code.

Algorithm 1: Mystery Function

```
Function  $F(n)$ :  
  If  $n = 0$  :  
     $\sqsubset$  Return  $(2, 1)$   
  Else  
     $b \leftarrow 1$   
    For  $i$  from 1 to  $n$   
       $\sqsubset$   $b \leftarrow 2b$   
     $(u, v) \leftarrow F(n - 1)$   
    Return  $(u + b, v \cdot b)$ 
```

(a) What are the results of $F(1)$, $F(2)$, $F(3)$, and $F(4)$?

Solution:

(b) What does the code do in general, when given input integer $n \geq 0$? Prove your assertion by induction on n .

Solution:

Problem 2. (12 points) *Making exact change*

In the country of Perfect Squares, all coins are in denominations that are perfect squares i.e. i^2 for some integer i . You need to buy a jacket whose price is an integer $n \geq 1$. The country is obsessed with being perfect and you can only buy an item if you pay the exact price. You happen to have exactly one coin of value i^2 for each integer $i \geq 2$. Additionally you have 4 coins of value 1 each. Use induction to show that you can pay the exact cost of the jacket using your coins.

Solution:

Problem 3. (12 points) *More induction practice*

Consider the following function f defined on the nonnegative integers.

$$f(0) = 3$$

$$f(1) = 4$$

$$f(n) = 3f(n-2) + 2f(n-1), \text{ for } n \geq 2$$

Prove by induction that $f(n) = (5 \cdot (-1)^n + 7 \cdot 3^n)/4$ for all integer $n \geq 0$.

Solution:

Problem 4. (12 points) *Growth of functions*

Arrange the following functions in order from the slowest growing function to the fastest growing function. Note that $\lg n = \log_2 n$.

$$n^{2/3} \quad n + \lg n \quad 2^{\sqrt{\lg n}} \quad (\lg n)^{\lg n}$$

Justify your answers. Specifically, if your order is of the form

$$f_1 \quad f_2 \quad f_3 \quad f_4,$$

you should establish $f_1 = O(f_2)$, $f_2 = O(f_3)$, and $f_3 = O(f_4)$. For each case, your justification can be in the form of a proof from first principles or a proof using limits, and can use any of the facts presented in the lecture or the text. (Hint: It may help to plot the functions and obtain an estimate of their relative growth rates. In some cases, it may also help to express the functions as a power of 2 and then compare.)

Solution:

Problem 5. ($2 \times 5 = 10$ points) *Properties of asymptotic notation*

Let $f(n)$, $g(n)$, and $h(n)$ be asymptotically positive and monotonically increasing functions.

- (a) Using the formal definition of the O and Ω notation, prove that if $f(n) = O(h(n))$ and $g(n)^2 = \Omega(h(n)^2)$, then $f(n) = O(g(n))$.

Solution:

- (b) Give distinct functions f and g satisfying both $f(n) = \Theta(g(n))$ and $3^{f(n)} = \Theta(3^{g(n)})$.
Give distinct functions f and g satisfying $f(n) = O(g(n))$ yet $3^{f(n)} \neq O(27^{g(n)})$.

Solution:

Problem 6. (12 points) *Determining the largest element in one list that is not present in another list*

We have learned that Mergesort sorts an array of n numbers in $O(n \log n)$ time and Binary Search determines if a given number is present in a sorted array of n numbers in $O(\log n)$ time.

Describe an $O(n \log n)$ time algorithm that takes as input two arrays A and B with n elements each (not necessarily sorted) and determines the largest number in B that is not in A . If all elements of B are in A , then return "All elements of B are in A ".

Your algorithm should use mergesort and binary search and should not use hash tables. Give your algorithm in pseudocode. Justify the running time of your algorithm.

Solution: