Homework 1

CS 4104 (Summer 2024)

Assigned on June 7, 2024.
Submit a PDF file containing your solutions on Canvas by 11:59pm on Friday June 14, 2024.

Instructions:

- The Honor Code applies to this homework with the following exception:
  - You are allowed to discuss possible algorithms and bounce ideas with your classmates. Do not discuss proofs of correctness, final answer or running time in detail with your classmate. You must write down your solution individually and independently. Do not send a written solution to your classmates for any reason whatsoever.
  - You are not allowed to consult sources other than your textbook, the slides on the course web page, your own class notes, the TA, and the instructor. **In particular, do not use a search engine or large language models such as ChatGPT**.
- Do not forget to typeset your solutions. Every mathematical expression must be typeset as a mathematical expression, e.g., the square of n must appear as n^2 and not as `n2`. You can use the LaTeX version of the homework problems to start entering your solutions. At the end of each problem are three commented lines that look like the example below. You can uncomment these lines and type in your solution within the curly braces.

```
% \solution{
%
% }
```

- Do not make any assumptions not stated in the problem. If you do make any assumptions, state them clearly, and explain why the assumption does not decrease the generality of your solution
- You must also provide a clear proof that your solution is correct (or a counter-example, where applicable). Type out all the statements you need to complete your proof. You must convince us that you can write out the complete proof. You will lose points if you work out some details of the proof in your head but do not type them out in your solution.
- If you are proposing an algorithm as the solution to a problem, keep the following in mind:
  - Describe your algorithms as clearly as possible. The style used in the text book is fine, as long as your description is not ambiguous. Explain your algorithm in words. A step-wise description is fine. *However, if you submit detailed pseudo-code without an explanation, we will not grade your solutions.*Do not describe your algorithms only for a specific example you may have worked out.
  - Make sure to state and prove the running time of your algorithm. You will only get partial credit if your analysis is not tight, i.e., if the bound you prove for your algorithm is not the best upper bound possible.
  - You will get partial credit if your algorithm is not the most efficient one that is possible to develop for the problem.
- In general for a graph problem, you may assume that the graph is stored in an adjacency list and that the input size is m + n, where n is the number of nodes and m is the number of edges in the graph. Therefore, a linear time graph algorithm will run in $O(m + n)$time.

Acknowledgement: Some of the questions in this Homework are inspired by/adopted from previous CS 4104 offerings at VT including by T. M. Murali and others.

Problem 1 (5 points) If a potential solution to the Stable Matching problem is a perfect matching but is not stable, then the number of rogue **couples must be at least two**. Just say "True" or "False". You do not have to provide any reasoning,

Problem 2 (10 points) In any input to the Stable Matching problem with n women and n men, what is the number of perfect matchings? Prove your answer.

Problem 3 (10 points) Prove the correctness of the following algorithm for finding the maximum element in an array.

```
def find_max(arr):
    max_val = arr[0]
    for i in range(1, len(arr)):
        if arr[i] > max_val:
            max_val = arr[i]
    return max_val
```

Problem 4 (5 points) Calculate the runtime of the following algorithm that checks if a number is prime (worst case).

```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
```

Problem 5 (5 Points) Prove that $3n^2 + 5n + 2$is $O(n^2)$.

Problem 6 (5 Points) Compare the growth rates of the following functions and order them from slowest to fastest:

$$f_1(n) = \log n, \quad f_2(n) = n, \quad f_3(n) = n \log n, \quad f_4(n) = n^2.$$

Problem 7 (10 Points) Prove that $4n^3 + 2n^2 + 7n + 1$ is $\Theta(n^3)$.

Extra Credit (10 Points) Determine the space complexity of the following algorithm that computes the sum of an array.

```
def array_sum(arr):
    total = 0
    for i in range(len(arr)):
        total += arr[i]
    return total
```