Homework 5

CS 4104 (Spring 2024)

Assigned on April 1, 2024.
Submit PDF solutions on Canvas by
11:59pm on April 8, 2024.

Instructions:

- The Honor Code applies to this homework with the following exception:
  - You can pair up with another student to solve the homework. Please form teams yourselves. Of course, you can ask the instructor for help if you cannot find a team-mate. You may choose to work alone.
  - You are allowed to discuss possible algorithms and bounce ideas with your team-mate. **Do not discuss proofs of correctness or running time in detail with your team-mate. You must write down your solution individually and independently. Do not send a written solution to your team-mate for any reason whatsoever.**
  - *In your solution, write down the name of the other member in your team. If you do not have a team-mate, please say so.*
  - Apart from your team-mate, you are not allowed to consult sources other than your textbook, the slides on the course web page, your own class notes, the TAs, and the instructor. In particular, do not use a search engine or any other online sources including ChatGPT or generative AI software of that ilk.
- **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 "n^2".* You can use the LaTeX version of the homework problems to start entering your solutions.
- 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 (the strategies are based on mistakes made by students over the years):
  - Describe your algorithms as clearly as possible. The style used in the 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 code or 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.

My team-mate is _____

Problem 1 (20 points) In this problem, you will analyse the worst-case running time of weighted interval scheduling *without* memoisation. Recall that we sorted the n jobs in increasing order of finish time and renumbered these jobs in this order, so that $f_i \leq f_{i+1}$, for all $1 \leq i < n$, where f_i is the finish time of job i. For every job j, we defined $p(j)$ to be the job with the largest index that finishes earlier than job j. Consider the input in Figure 6.4 on page 256 of your textbook. Here all jobs have weight 1 and $p(j) = j - 2$, for all $3 \leq j \leq n$ and $p(1) = p(2) = 0$. Let $T(n)$ be the running time of the dynamic programming algorithm *without memoisation* for this particular input. As we discussed in class, we can write down the following recurrence:

$$T(n) = T(n-1) + T(n-2), n > 2$$
$$T(2) = T(1) = 1$$

Prove an *exponential lower* bound on $T(n)$. Specifically, prove that $T(n) \geq 1.5^{n-2}$, for all $n \geq 1$.

Problem 2 (35 points) Solve exercise 1 in Chapter 6 (pages 312–313) of your textbook.

Problem 3 (45 points) After graduation, you start a job at a company that is building a space elevator! People travelling to space in the elevator need to eat. Therefore, your company is planning to open a series of restaurants along the elevator. There are n potential locations for these restaurants. Starting from the beginning of the space elevator, these locations are in miles and in increasing order, $m_1, m_2, m_3, \ldots m_{n-1}, m_n$. Your company wants to minimize the costs of constructing and maintaining these restaurants as well as maximising its profits. It devises the following rules:

(i) At each location, it can open at most one restaurant.

(ii) For each $1 \leq i \leq n$, the expected profit from opening a restaurant at location i is $p_i > 0$.

(iii) Any two restaurants your company builds should be at least k miles apart, where k is a positive integer.

Since the company hired you because you have taken CS 4104, it asks you to devise an efficient algorithm to decide where to build the restaurants so as to compute the maximum total profit subject to the given rules. Use your advanced knowledge of algorithms to impress your bosses and peers and collect a hefty Christmas bonus!