

Homework 6

CS 4104 (Spring 2024)

Assigned on April 10, 2024.

Submit PDF solutions on Canvas by the
11:59pm on April 17, 2024. **There will be no extensions.**

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 “ $n^{\wedge}2$ ”. You can use the L^AT_EX 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.

Additional notes.

- For some of the problems in this homework, you will reduce the given problem to one of the flow-related problems we solved in class. Make sure you describe the reduction completely. For example, if you reduce a problem to the maximum network flow problem itself, specify clearly how will convert an input to the original problem into a flow network (nodes, source and sink, edges, directions, and edge capacities). Do not forget to prove the correctness of your solution in both directions, as we did in class (lest you transmogrify Calvin-sized Hobbes into a dinosaur).
- For some of the problems in this homework, you do not have to reduce the problem “all the way” to computing the maximum flow in a network. A reduction to maximum bipartite matching or to minimum cut may suffice. You can then use the polynomial time algorithms we have developed for these problems to solve the original problem. In other words, you do not have to also describe how to reduce maximum bipartite match matching or minimum cut to the maximum flow problem.

Problem 1 (25 points) Suppose you have a potential solution to a maximum-flow problem, i.e., you have a flow network $G = (V, E)$ (with s and t and edge capacities, defined, of course) and an assignment $f : E \rightarrow \mathbb{R}^+$ that satisfies the capacity and conservation conditions. Develop an efficient algorithm to determine if f is indeed a maximum flow. *Hint:* Consider the algorithm for computing a minimum cut if you know for sure that f is a maximum flow, which we discussed in class.

Problem 2 (35 points) Given a flow network G , suppose f is an s - t flow with positive value, i.e., $\nu(f) > 0$. Consider the subgraph G' of G that only consists of edges that carry positive flow, i.e., an edge e is in G' if and only if $f(e) > 0$. Prove that there must be a simple s - t path in G' . Note that G' is not the residual graph. *Hint:* Here is the beginning of a proof by contradiction. Suppose there is no simple s - t path in G' . Let A be the set of nodes reachable from s in G' . If $t \in A$, then clearly t is a member of A . Hence, $(A, V - A)$ is a s - t cut.

Problem 3 (40 points) Solve exercise 6 in Chapter 7 (pages 416–417) of your textbook.