Please enter your name and uID below.

Name:

uID:

Collaborators, if any, and how you collaborated:

Submission notes

- Due at 11:59 pm (midnight) on WEDNESDAY, Nov 23rd.
- Solutions must be typeset using one of the template files. For each problem, your answer must fit in the space provided (e.g. not spill onto the next page) *without* space-saving tricks like font/margin/line spacing changes.
- Upload a PDF version of your completed problem set to Gradescope.
- Teaching staff reserve the right to request original source/tex files during the grading process, so please retain these until an assignment has been returned.
- Please remember that for this problem set, you are allowed to collaborate in detail with your peers, as long as you cite them. However, you must write up your own solution, alone, from memory. If you do collaborate with other students in this way, you must identify the students and describe the nature of the collaboration. You are not allowed to create a group solution, and all work that you hand in must be written in your own words. Do not base your solution on any other written solution, regardless of the source.

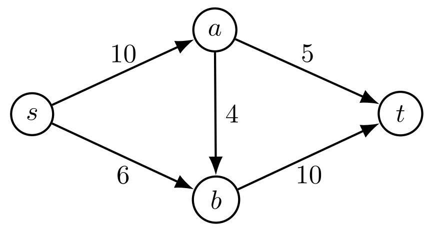
1. (Commute Times, 30pts)

   Describe and analyze an algorithm to solve the PS9 Programming problem. Your algorithm should take as input an undirected graph G (represented as an adjacency list) with weights $w : E \to \mathbb{R}$, where the vertices of G represent intersections and $w(u, v)$ gives the length of an *existing* road between intersections u and v. You may also assume there is some global function $dist(u, v)$ which returns the physical distance between u and v: $dist$ tells you how long a new road would be, if built. Your algorithm should return the smallest possible *new* commute time for G, w if you were able to add one road to G.

   As with previous problem sets, your algorithm description should clearly and concisely describe the details of how you would solve this problem. Your analysis should give the runtime of the algorithm, and an explanation of how you arrived at that runtime. For full credit, your algorithm *must run in* ***less than*** $O(n^5)$ time (though a correct but slow algorithm is better than a fast but incorrect algorithm).

   Hint: there is a straightforward brute-force $O(n^5)$ *algorithm, you should try to think of that first. Once you have that algorithm, you should observe that you are re-calculating the same value many many times. How can you speed it up by getting rid of some redundant calculations?*

2. (Feasible Flows, 40pts) Let G be the following flow network with source s, target t, and capacities indicated on each edge.



   a. Let f be the flow in G with values $f(s \to a) = (a \to t) = 5$, and 0 for every other edge. $|f| = 5$, and f saturates the edge $a \to t$. Describe a feasible flow f' for G which saturates *exactly* 3 edges and has value $|f'| = 10$.

   b. Describe a feasible flow f'' for G with value $|f''| = 15$.

   c. Describe the residual graph for G_f by listing its non-zero edge capacities or drawing a small, clear, picture.

   d. Describe two feasible flows for the residual graph G_f, one with value $|f'| - |f| = 5$ and the other with value $|f''| - |f| = 10$.

   e. Now, consider *any* f, f', G[1]. Let f and f' be two feasible (s,t)-flows in some flow network G, such that $|f'| > |f|$. Prove that there is a feasible (s,t)-flow with value $|f'| - |f|$ in the residual network G_f.

      Prove this by showing exactly how you could construct a new flow in the residual graph (give the value for each edge of the new flow), given f' and f, and argue that this flow would be feasible and also have the desired value.

[1] You may assume G is reduced.