CSE 551 Midterm Test 2 Solutions
02 November 2021, 1:30-2:45 p.m. in Room

Last Name	SAMPLE	ASU ID	3336669990
First Name(s)	Ima	Exam #	1

I attest that I am the student whose name and number is listed on this paper.

I have read the instructions given on this page.

Signature:

Regrading of Midterms

If you believe that your grade has not been recorded correctly, return the entire paper to the instructor with a short note indicating what you believe to be the error. Other than for that reason, test grades are almost never changed. If you believe that you did not receive the proper credit, first **read these sample solutions carefully** to see if you can understand the answer to your concern. If that does not resolve it, write a clear explanation of why you believe the grade is in error and submit that, along with the entire test paper, to the instructor. Please do not discuss in your explanation how your solution is like that of another student, as FERPA legislation makes it impossible for me to discuss one student's work with another. Please take into account that many papers were graded, and it is impossible to change the grade on one paper without giving every other student the same opportunity. If you nevertheless want the paper regraded, be advised that the entire paper will be regraded and the grade may go up, stay the same, or go down. The new grade will be final. It is a violation of the Academic Integrity Policy to request a grade change simply because you need or want a higher grade.

If you require a clarification of the sample solutions (not a grade change or review as discussed above), ask in the office hours of the TA or instructor. You will be asked whether you have read the sample solution and to indicate what precisely is unclear to you about it, so **read these sample solutions carefully** first. Under no circumstances can anyone change a grade other than the instructor, so do not ask a TA to do so.

Grade review requests, whether submitted as described above or not, will not be considered if received after the start of class on 16 November 2021. If you cannot get it to me in class, you can ask the staff in the SCAI Main Office to put it in my mailbox. I need the original, not a scan or photo.

Instructions

Do not open the exam until 1:30 p.m., or you are instructed to do so. You have 75 minutes to complete the exam. When the time is over, stop writing. No books, notes, electronic devices, or other aids are permitted. Turn all devices OFF; put them in your backpack or with your books. *This includes phones, watches, etc.* Place backpacks, notes, texts, etc. away from your work space, e.g., at the side or front walls. You only need your pens. Write all answers on the examination paper itself. Give one answer to each question; multiple answers to a single question will result in lower grades. Your answer should be both precise and concise; do not include unnecessary information. BUDGET YOUR TIME WELL! SHOW ALL WORK!

YOU WILL NOT BE PERMITTED TO START THE TEST IF YOU ARRIVE AFTER 2.15 P.M. ONCE STARTED, YOU MUST REMAIN IN YOUR SEAT UNTIL AT LEAST 2:15 P.M. IF YOU FINISH BETWEEN 2:15 P.M. AND 2:40 P.M., YOU MAY — QUIETLY — HAND IN YOUR PAPER, COLLECT YOUR BELONGINGS, AND LEAVE. IF YOU HAVE NOT LEFT BY 2:40 P.M., REMAIN IN YOUR SEAT UNTIL YOUR PAPER IS COLLECTED BY THE TA OR PROFESSOR AND THEN COLLECT YOUR BELONGINGS AND LEAVE THE TESTING ROOM. DO NOT write anything on your test paper after 2:45 p.m. Do NOT bring your paper to the TA or professor Do NOT chat with other students until you are outside. Failure to follow this protocol may result in a grade of 0 on the test.

Sample, Ima					
Question	Mark	Out Of	Question	Mark	Out Of
1		5	2		10
3		15	4		10
5		10	6		10
Total		60			

Question 1. [5 marks] At the state fair, there is a game with n balloons that are placed adjacent to each other in a line. In this game, you must shoot all the balloons. Each balloon $b \in \{1, 2, \ldots, n\}$ has an integer weight $v(b)$; weights may be positive, negative, or zero. Shooting balloon b earns you nothing if $b = 1$ or $b = n$; otherwise it earns an award of $v(b-1)v(b)v(b+1)$. After a balloon is shot, its former neighbours become adjacent, and the balloons are renumbered to be $1, 2, \ldots, n-1$, keeping their initial weights. Your goal is to earn the maximum total award after shooting all balloons.

For the statements that follow, there are ten balloons, of which at most five have weight 0. Mark each box provided with $\surd$ if the statement is correct. If incorrect, leave it blank or mark it with an $\times$.

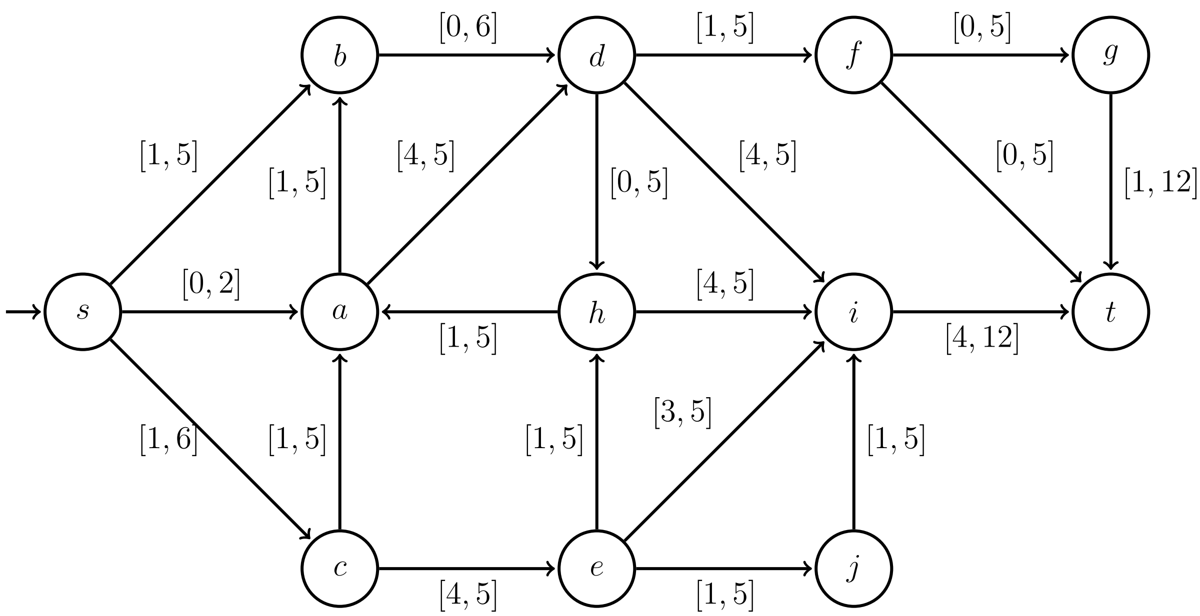
- [$\surd$] It is possible that the maximum total award is positive.
- [$\times$] It is possible that the maximum total award is negative. You can always shoot balloons at the ends to get total award 0, so it cannot be negative.
- [$\surd$] It is possible that the maximum total award is zero. For example, if all balloons have negative weight.
- [$\times$] To get maximum total award, we can always shoot all zero weight balloons first. In the first version of these solutions, this was wrongly indicated as correct. But it is not. Using an example by Robert Nordman from the Canvas discussion board, try $110x110000$ where $x \leq -3$. Shooting all 0-weight balloons first leaves $11x11$, and the total award cannot exceed 0. Instead shoot the x first, then all 0-weight balloons, to leave 1111, then shoot a middle balloon (twice) to get total award 2. Therefore the given statement is incorrect.
- [$\times$] To get maximum total award, we can always shoot all zero weight balloons last. Try 0101010101.

Question 2. [10 marks] Our input is a directed graph G with n nodes, source s, sink t, and integer lengths on its directed edges. We are guaranteed that G contains at least one s,t-path. Edge lengths may be positive or negative, and there may be negative cycles. I am to find an s,t-path whose total length is as small as possible (a path can traverse each directed edge at most once). Mark each box provided with $\surd$ if the assertion is definitely true for this scenario. Otherwise, leave it blank or mark it with an $\times$.

- [$\times$] There must be a polynomial time algorithm that always finds an s,t-path in G whose total length is minimum.

- √ There is an exponential time algorithm that always finds an s, t-path in G whose total length is minimum. There are "only" exponentially many s, t-paths, so list them all and pick a minimum.
- × It may be that no s, t-path in G whose total length is minimum exists, so no algorithm can find one. There are fiinitely many s, t-paths, so there is a minimum.
- × Dijkstra's algorithm always finds an s, t-path in G whose total length is minimum.
- × Dijkstra's algorithm always finds an s, t-path in G whose total length is minimum, if we first add a large number to each edge length so that they are all positive. The question asks about G, not the modified graph.
- √ Dijkstra's algorithm always finds an s, t-path in G, but its total length may not be the minimum.
- × The Bellman-Ford algorithm always finds an s, t-path in G whose total length is minimum.
- × The Bellman-Ford algorithm always finds an s, t-path in G whose total length is minimum, if we first remove all edges that appear in any negative cycle. You might remove all of the edges if you do this!
- × The Bellman-Ford algorithm always finds an s, t-path in G whose total length is minimum, if we only run the algorithm for $n - 1$ iterations.
- × The Ford-Fulkerson algorithm always finds an s, t-path in G whose total length is minimum.

Question 3. [15 marks] A directed graph G is given. It has source s, sink t, and both a lower and an upper bound on the capacity of each edge in the form $[lower, upper]$.



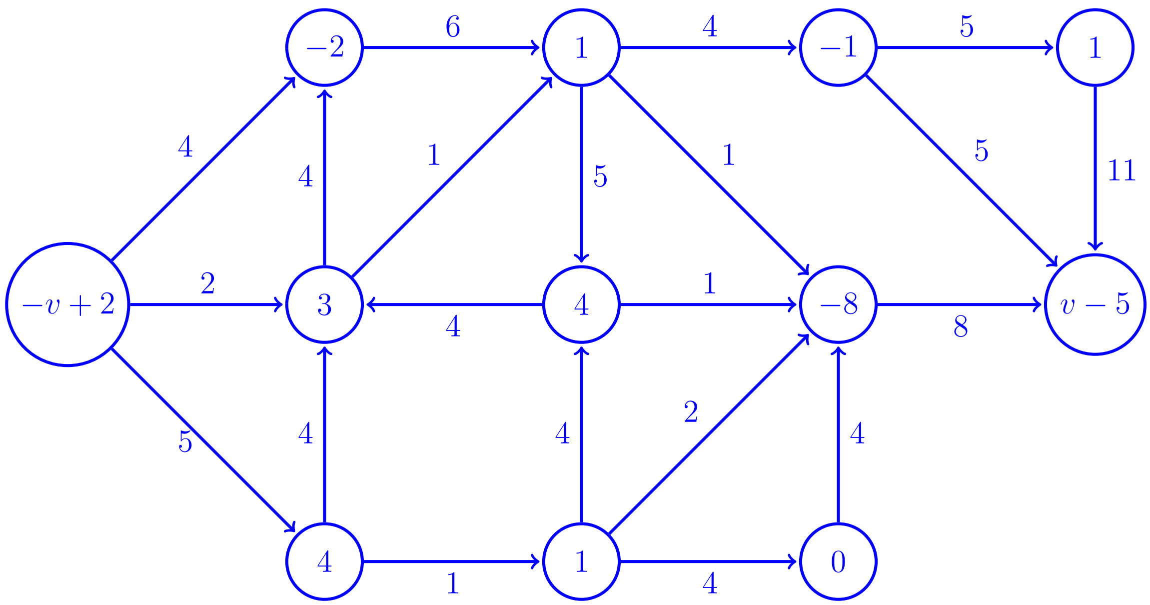
(A copy of this picture is on the second rough work page, in case you need a clean copy.)

Fill in the each of the boxes with a nonnegative integer.

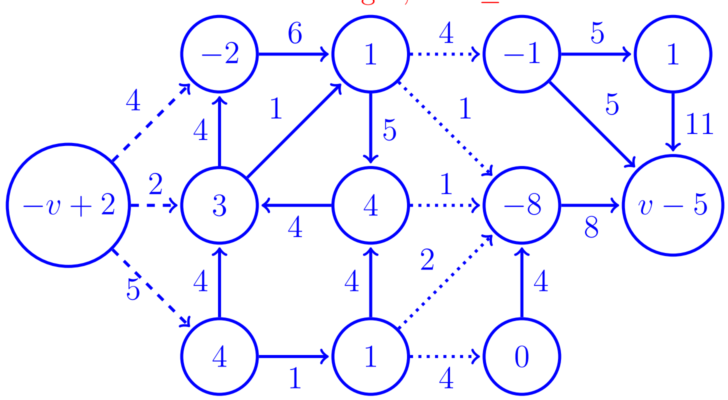
The value of a feasible s, t-flow is at least $\boxed{13}$ and at most $\boxed{13}$. In a feasible s, t-flow, the flow on edge (f, t) is at least $\boxed{0}$ and at most $\boxed{0}$. In a feasible s, t-flow, the flow on edge (i, t) is at least $\boxed{12}$ and at most $\boxed{12}$. In a feasible s, t-flow, the flow on edge (e, i) is at least $\boxed{3}$ and at most $\boxed{3}$.

In a feasible s,t-flow, the flow on edge (h,a) is at least $\boxed{2}$ and at most $\boxed{2}$. In a feasible s,t-flow, the flow entering node d is at least $\boxed{10}$, the flow entering node f is at least $\boxed{1}$, the flow entering node h is at least $\boxed{6}$, the flow entering node b is at least $\boxed{6}$, and the flow entering node e is at least $\boxed{5}$.

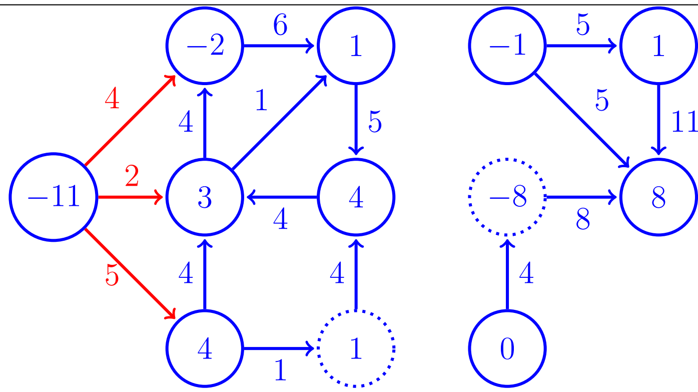
You did not need to explain. But can I convince you that these are all correct? To verify, first convert to a circulation problem with demands by eliminating the lower bounds.



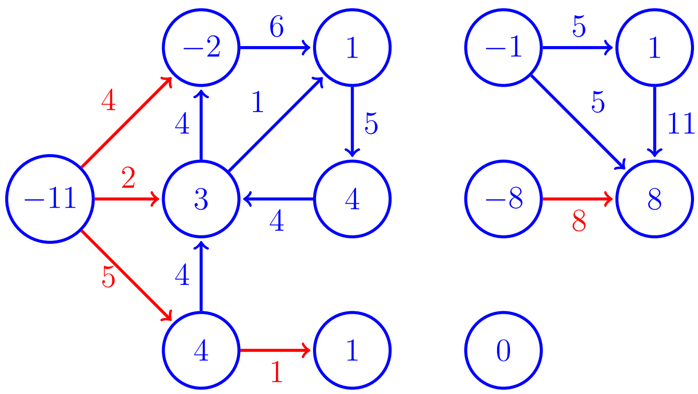
In the following the dashed cut says that $0 - 11 \leq -v + 2$ so $v \leq 13$. The dotted cut then has total demand $-v + 13$ on the left and $v - 13$ on the right, so $v \geq 13$.



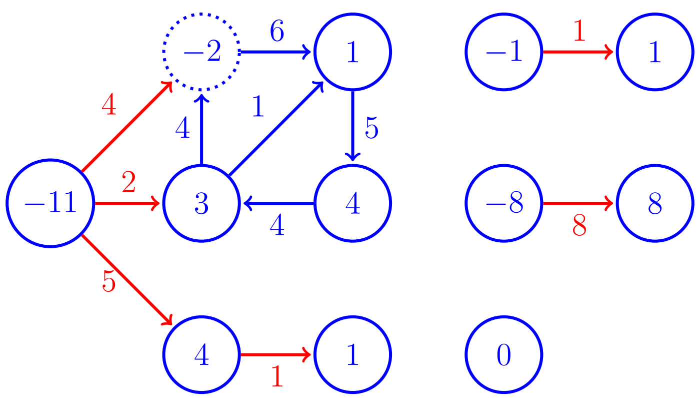
No edge can carry a flow across the dotted cut. All three dashed edges are saturated.



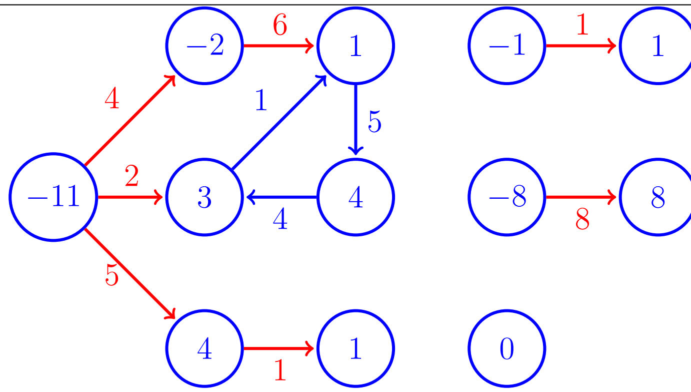
Look at the two dotted nodes. Compute inflow minus outflow to equal demand:



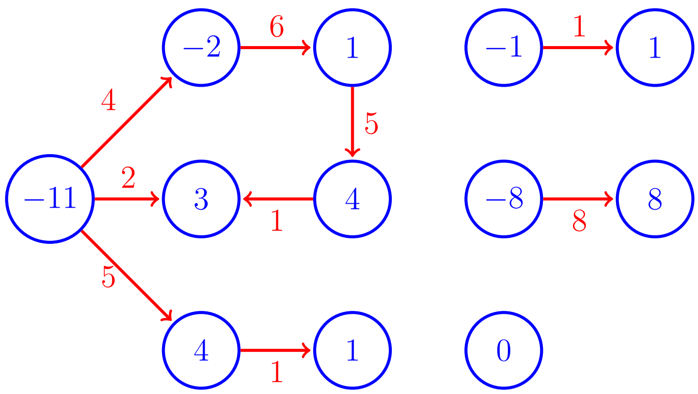
Continuing,



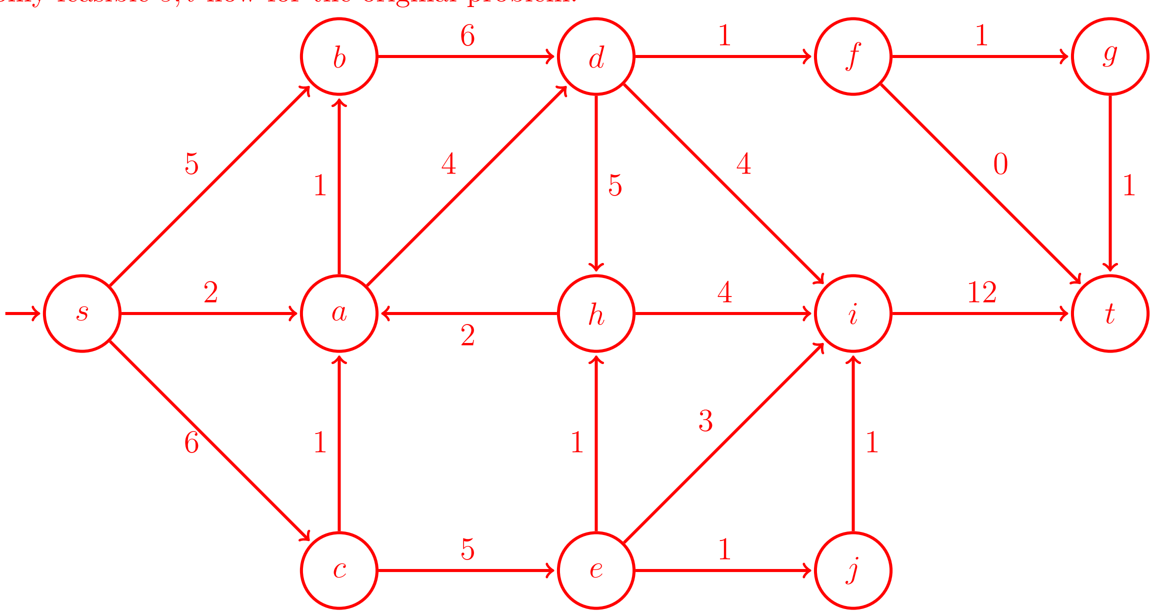
Consider flows at the dotted node to get



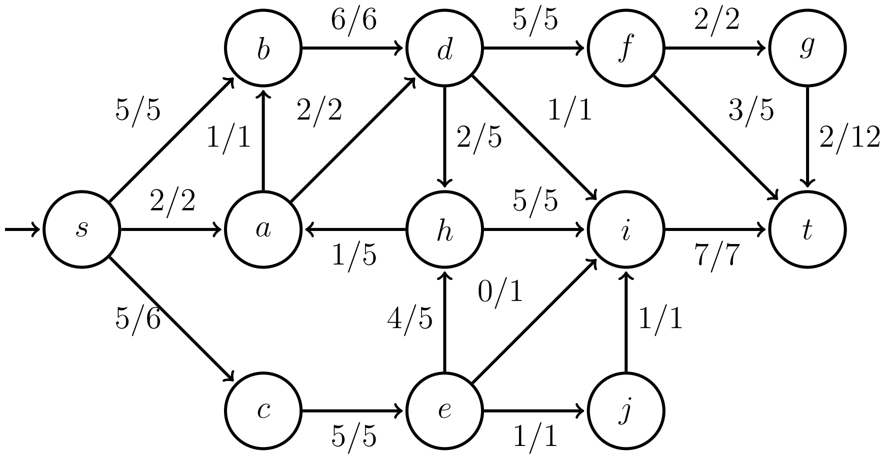
Finishing up:



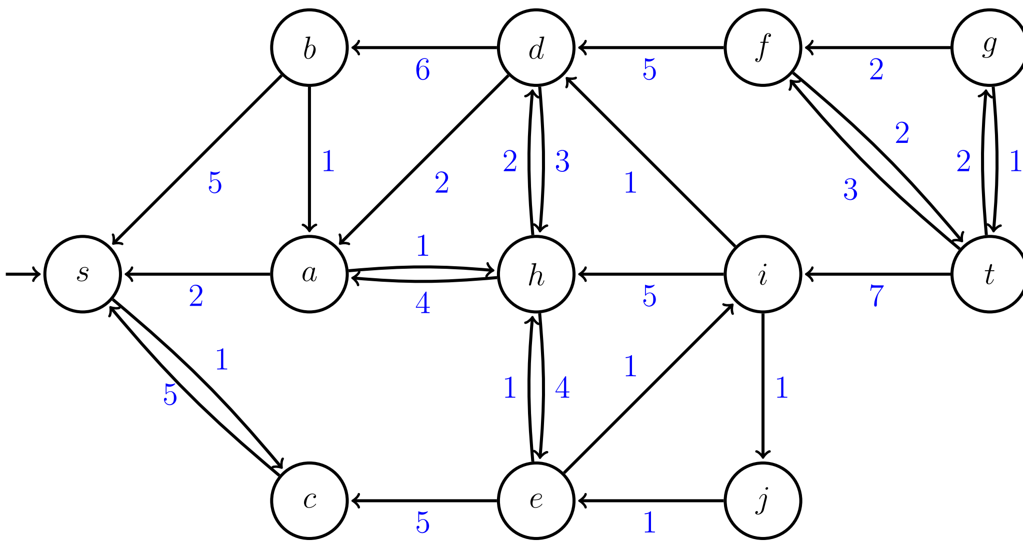
This is the only feasible circulation with demands. Add the lower bounds back to the flows to get the only feasible s,t-flow for the original problem.



Question 4. [10 marks] A directed graph G is given. It has source s, sink t, and a capacity $c(e)$ of each edge e. An s,t-flow f is also given by labelling each directed edge e by "$f(e)/c(e)$."



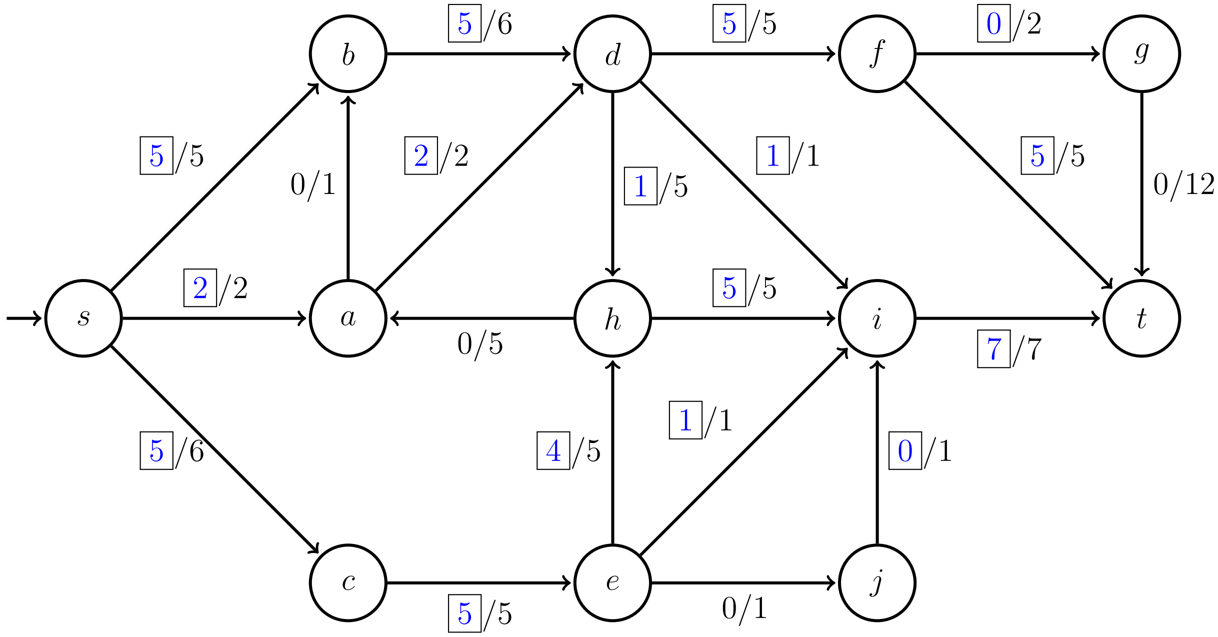
Show the edges and residual capacities for the residual graph G_f on the following picture:



As expected, there is no augmenting s, t-path in this residual graph.

Find an s, t-flow in which edges (a, b), (h, a), (e, j), and (g, t) each have flow 0, but the s, t-flow has the same value as f (fill in the boxes with the flow):

To find one, you can augment along the directed cycles (a, h, d, b) with bottleneck 1, (e, i, j) with bottleneck 1, and (g, f, t) with bottleneck 2 to get the following solution.



(A copy of this picture is on the second rough work page, in case you need a clean copy.)

Question 5. [10 marks] For a directed graph G with source s and sink t, recall that an s, t-cutset is a set of directed edges whose removal separates t from s. Menger's Theorem states: *The maximum number of edge disjoint s, t-paths is equal to the minimum number of edges in an s, t-cutset.* My friend is confused – he thinks it says: *The maximum number of edge disjoint s, t-cutsets is equal to the minimum number of edges in an s, t-path.* First state whether my friend's statement is true or false. Then justify your answer carefully (i.e., prove that it is true or give a counterexample).

This is not max-flow/min-cut in disguise. It is quite different – even though the basic proof strategy is the same (but easier).

TRUE. Let γ be the maximum number of edge-disjoint s, t-cutsets and δ be the length of a shortest s, t-path (in terms of number of edges). Now we use the fact that every s, t-cutset contains at least one edge from every s, t-path. It follows that $\gamma \leq \delta$. To show equality, do a BFS to label all nodes with their distance (in terms of number of edges) from s (then t will get label δ). Form an s, t-cutset C_i by including all edges from a node with label $i - 1$ to a node with label i for $1 \leq i \leq \delta$. Then $C_1, \ldots, C_\delta$ is a set of edge-disjoint s, t-cutsets, so $\gamma \geq \delta$.

Many students answered a different question. This question does not involve capacities or flows, it does not involve edge-disjoint paths or the size of a mincut. It does not involve reversing any edges or swapping s and t. Read it carefully: It says that my friend thinks that the maximum number of edge disjoint s, t-cutsets is equal to the minimum number of edges in an s, t-path.

Question 6. [10 marks] We are given a directed graph G with source s and sink t. Each edge has a length that can be positive, zero, or negative, but every directed cycle in G has positive total cost.

Devise the most efficient algorithm that you can that, given such a G, s, t, computes the <u>number</u> of distinct s, t-directed paths of minimum total length. Ensure that the main steps of your method are given completely and correctly, but do not explain implementation-level details.

There can be exponentially many shortest paths, so any attempt to list them all must yield an exponential time algorithm.

For reference, here is a version of the Bellman-Ford algorithm.

Shortest-Path(G, t)
  foreach node $v \in V$
    $M[0, v] = \infty$
  $M[0, t] = 0$
  for $i = 1$ to $n - 1$
    foreach node $v \in V$
      $M[i, v] = M[i - 1, v]$
      foreach edge $(v, w) \in E$
        $M[i, v] = \min(M[i, v], M[i - 1, w] + c_{vw})$

Adapt a version of the Bellman-Ford algorithm. Compute and maintain certain values for each $v \in V$:

- `dist`$[v]$ is the length of a shortest v, t-path found so far,
- `nump`$[v]$ is the number of them

and for each iteration $0 \leq i < n$ and each $v \in V$:

- `levdist`$[i, v]$ is the length of a shortest v, t-walk on exactly i edges,
- `levnump`$[i, v]$ is the number of them.

The proposed method follows Bellman-Ford with extra bookkeeping. Although we consider *walks* (allowing repeated nodes) rather than paths, any shortest v, t-walk must be a path, because if it contained a cycle (which the problem states must have positive cost), we could remove the edges of the cycle to get a shorter v, t-walk.

Count-Shortest-Paths(G, t)
  foreach node $v \in V$
    `levdist`$[0, v] = \infty$; `dist`$[v] = \infty$; `nump`$[v] = 0$; `levnump`$[0, v] = 0$
  `levdist`$[0, t] = 0$; `dist`$[t] = 0$; `nump`$[t] = 1$; `levnump`$[0, t] = 1$
  for $i = 1$ to $n - 1$
    foreach node $v \in V$
      `levdist`$[i, v] = \infty$; `levnump`$[i, v] = 0$
      foreach edge $(v, w) \in E$
        if `levdist`$[i - 1, w] + c_{vw} <$ `levdist`$[i, v]$ then
          `levdist`$[i, v] =$ `levdist`$[i - 1, w] + c_{vw}$
          `levnump`$[i, v] =$ `levnump`$[i - 1, w]$
        else if `levdist`$[i - 1, w] + c_{vw} =$ `levdist`$[i, v]$ then
          `levnump`$[i, v] =$ `levnump`$[i, v] +$ `levnump`$[i - 1, w]$
      if `levdist`$[i, v] <$ `dist`$[v]$ then {`dist`$[v] =$ `levdist`$[i, v]$; `nump`$[v] =$ `levnump`$[i, v]$}
      else if `levdist`$[i, v] =$ `dist`$[v]$ then {`nump`$[v] =$ `nump`$[v] +$ `levnump`$[i, v]$}
  Result: There are `nump`$[s]$ s, t-paths of minimum length `dist`$[s]$

A precise English language description would be fine as well.

You could try to run the original Bellman-Ford and trace back through the $M[]$ table to compute the number of s, t-paths, but this can be tricky. How can you be sure you do not count a path more than once?

You could try to modify Dijkstra's algorithm. But how do you deal with negative length edges?