NETWORK OPTIMIZATION

Megan Bryant

mrbryant@email.wm.edu

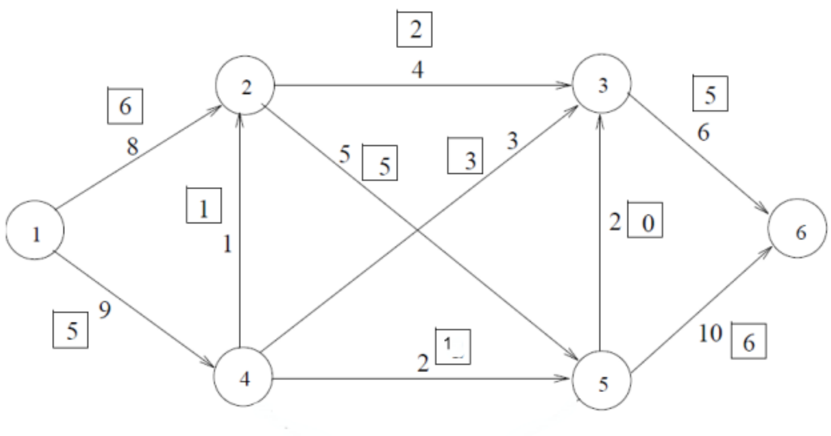
Department of Mathematics, The College of William and Mary, P.O. Box 8795, Williamsburg, VA 23187

Executive Summary

Homework 5.
Due date: March 19th, at the start of class
Collaborators:

1

The following figure gives an input to the maximum flow problem, and a feasible flow that has been computed by the Ford-Fulkerson algorithm (flow is given in box).



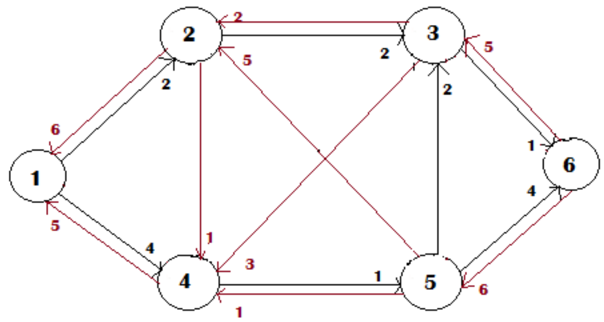
a.) *Finish the execution of this algorithm to produce a maximal flow.*

Ford Fulkerson Maximum Flow Algorithm (Augmenting Path)

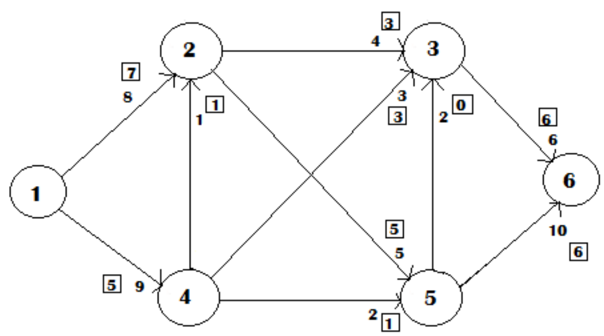
---

begin
  $x := 0$;
  while $G(x)$ contains a directed path from node s to node t **do**
  begin
    identify an augmenting path P from node s to node t;
    $\delta := \min\{r_{ij} : (i,j) \in P\}$;
    augment δ units of flow along P and update $G(x)$;
  end;
end;

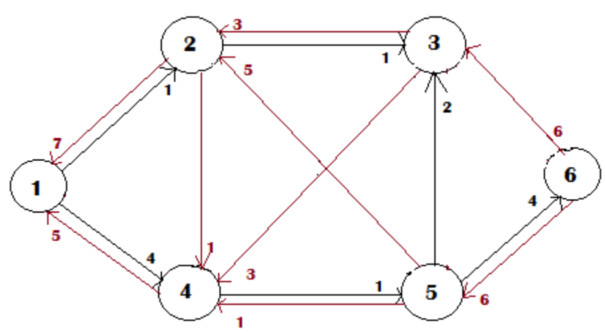
First, we will generate the residual graph of the initial feasible solution.



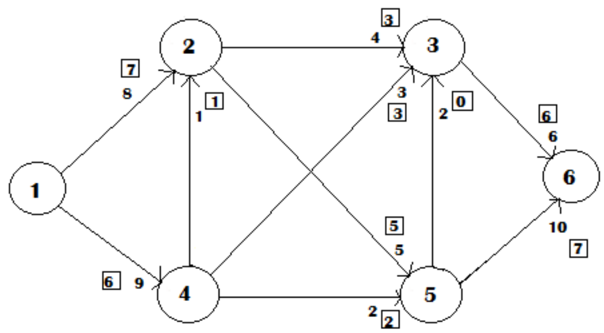
The nodes reachable from 1 are $2, 3, 4, 5, 6$. We can push a flow of 1 along path $1, 2, 3, 6$. The updated feasible solution is



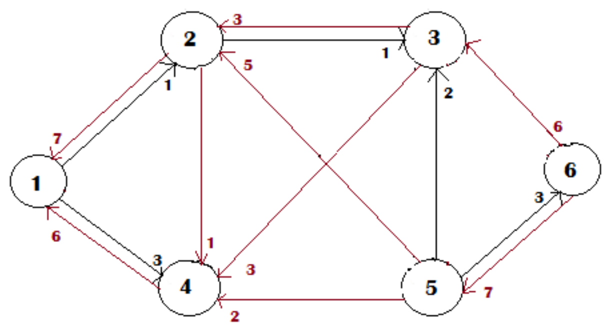
The corresponding residual graph is



We see that the nodes reachable from 1 are now $2, 3, 4, 5, 6$. Thus we haven't reached the max flow. Further, we see that we can push an additional one unit of flow along the path $1, 4, 5, 6$. The updated feasible solution is



The corresponding residual graph is



We see that node 6 is no longer reachable from node 1. As such, we conclude that we have an optimal solution to the maximum flow problem. The value of the solution is 13.

b.) *Use the Ford-Fulkerson algorithm to produce a cut of minimum capacity.*

According to the Ford Fulkerson algorithm, we should examine the arcs in the maximum flow residual graph with zero capacity. Therefore, we can see that the minimum S-T cut is $S = \{1, 2, 3, 4\}$ and $T = \{5, 6\}$. The capacity of this cut is the sum of the capacities of the cut edges: $(2, 5), (3, 6)$, and $(4, 5)$.

$$Cut = 6 + 2 + 5 = 13$$

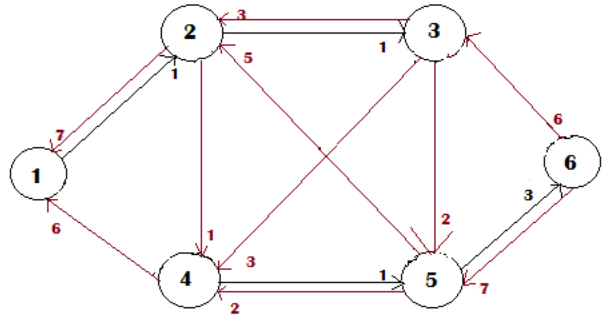
.

c.) *What is the value of the flow that you have found? What is the capacity of the cut that you have found?*

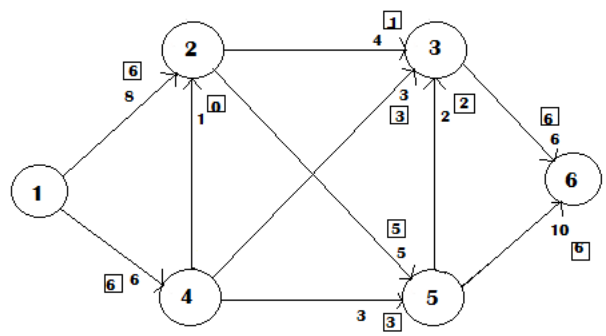
The value of the flow and the capacity of the cut are both 13.

d.) *Suppose now that the capacity of edge (1,4) decreases from 9 to 6 and the capacity of edge (4,5) increases from 2 to 3. Is the flow that you have found still optimal? If your answer is yes, provide evidence for your claim. If your answer is no, show how to augment the flow. You are not allowed to run the algorithm from scratch.*

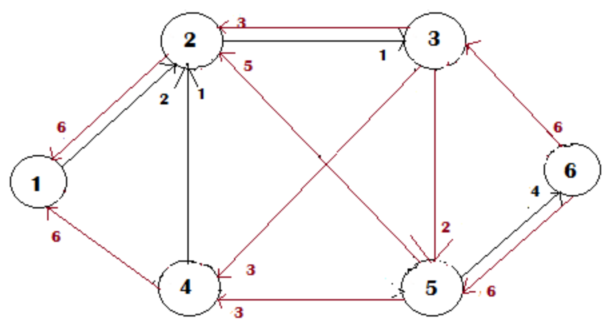
If the capacity on edge $(1, 4)$ decreases from 9 to 6, then the capacity of the edge would equal the flow in the optimal solution, which would not cause us to adjust our solution. If the capacity on edge $(4, 5)$ increased from 2 to 3, then we have a possible additional unit of flow available in the final solution. The corresponding residual graph is



We see that a viable augmenting path exists from $1, 2, 4, 5, 6$, so we will increase the flow along this path by 1. The updated feasible solution is



The corresponding residual graph is



We now have a maximal flow of value 12. The corresponding $S-T$ cut is $S = \{1, 2, 3\}$ and $T = \{4, 5, 6\}$ with a capacity 12.

2

Suppose that after solving a maximum flow problem, you realize that you underestimated the capacity of arc (i, j) by k units. Use the max flow-min cut theorem to show that the value of the new maximum flow is at most k more than the value of the old maximum flow, and conclude that the Ford-Fulkerson algorithm can reoptimize the flow using at most k iterations, and hence, at most $O(km)$ time. Give an example that shows *that in the worst case, you really need k iterations to reoptimize, no matter how you choose your augmenting paths.*

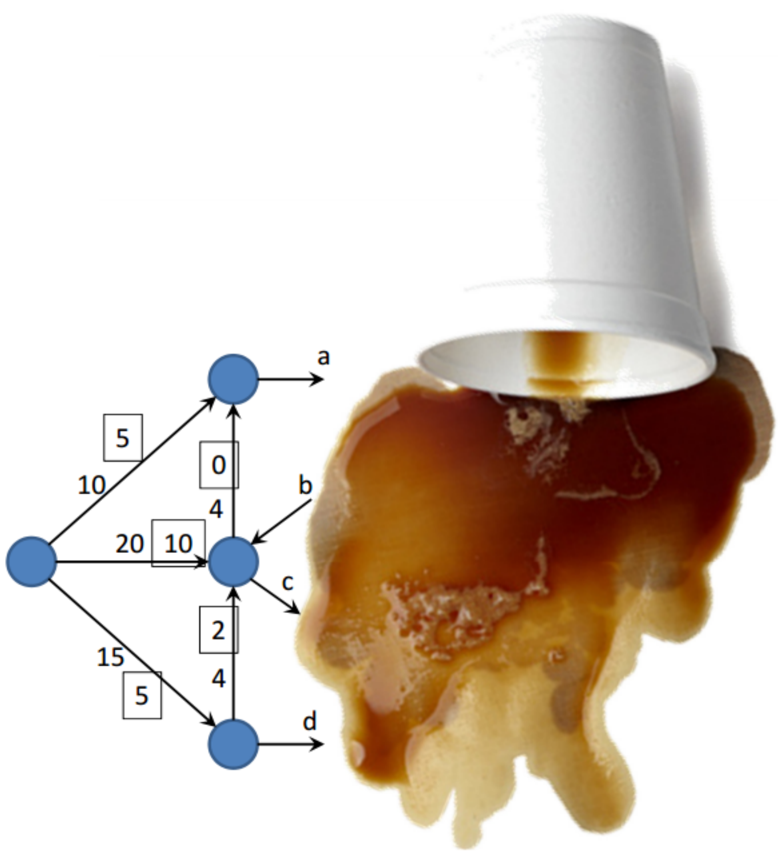
Max Flow-Min Cut Theorem: The maximum value of the flow from a source node s to a sink node t in a capacitated network equals the minimum capacity among all s-t cuts.

We know that since the capacities in this problem are integer, then the value of the flow must go up at each iteration by at least one unit. Now, we know that for the original optimization, the Ford-Fulkerson algorithm solves the original maximum flow problem in $O(nmU)$ time where $U = \max_{(v,w)\in A} u_{vw}$.

Suppose that capacity of the arc (i, j) is now the maximum among all arcs. Then the maximum capacity can increase by at most k units. Thus, $U' \leq U + k$ and the maximum value of the flow from a source node s to a sink node t can increase by at most k. Now, with regards to running time, we see that since the capacity of the cut $(s, N\ \{s\})$ is kU, the value of the maximum flow is at most kU and there are at most $O(kU)$ iterations.

3

Your friend is taking a network optimization course at a different university. She had to solve a similar problem to the one that you solved in problem 1. She asks you to check if her solution is correct. Being a very helpful, but also somewhat clumsy friend, you walk over to her desk with the intention of looking at her homework, but trip over your own shoe laces and spill your coffee on her papers. Below is all that remains of her flow graph.



a.) *Let x_{ij} denote the flow on arc (i, j) in your friend's solution. In the graph, there are four arcs which partially disappear into the coffee mess, labelled a, b, c, d. What is $x(a) - x(b) + x(c) + x(d)$ equal to? Justify* *your answer.*

We know that the flow on any arc leaving a node i is equal to the sum of the flow on the incoming arcs to i minus the sum of the outgoing flow on any other arcs leaving node i. Therefore, we have the following arc flows

$$\begin{aligned} a &= 5 + 0 = 5 \\ b &= c - (10 + 2) = c - 12 \\ c &= 10 + 2 + b = 12 + b \\ d &= 5 - 2 = 3 \end{aligned}$$

When calculating the requested sum, we should leave c in terms of c. Therefore, we have the following:

$$\begin{aligned} x(a) - x(b) + x(c) + x(d) &= 5 - (c - 12) + c + 3 \\ &= 5 - c + 12 - c + 3 \\ &= 20 \end{aligned}$$

This is the expected flow since the flow out of the source node is equal to $5 + 10 + 5 = 20$. Thus, the conservation of flow constraint is satisfied.

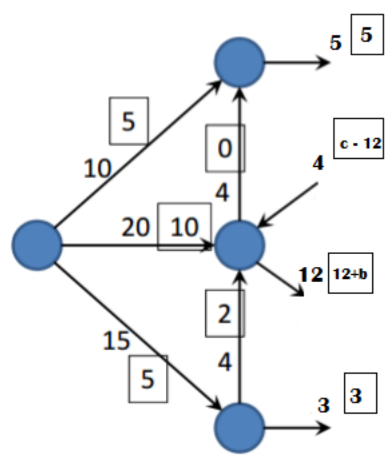
b.) *By going back to the original homework problem, you find out that the capacity of the arcs a, b, c, d is as follows: $u(a) = 5, u(b) = 4, u(c) = 12, u(d) = 3$.*

Based on this information, and the answer that you gave in ***a.)****, explain which (if any) of the following three statements cannot be true:*

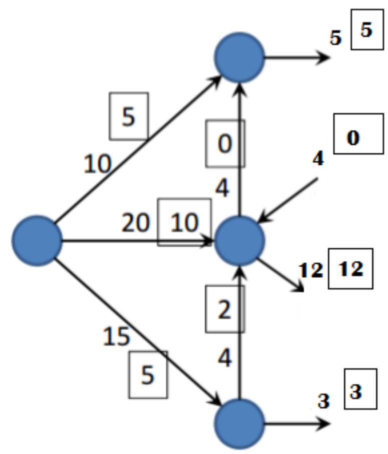
1. The flow x that your friend found is feasible, but not optimal. **True**
2. The flow x that your friend found is optimal. **False**
3. The flow x that your friend found is infeasible. **False**

Justify your answer

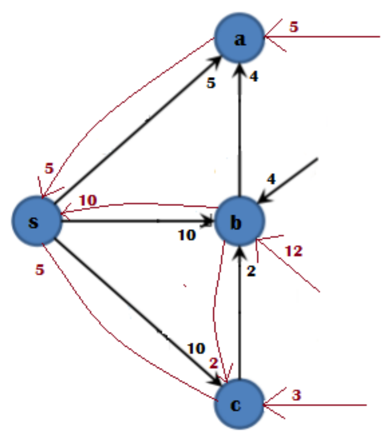
Based on our previous answer, the solution we have is feasible and thus statement 3 is not true and we have the following feasible solution



Now, if we are assuming basic competence of our friend, then we can assume that the flows on the arcs b and c must not exceed the capacity. Therefore, we know that $x(c) = 12$ and $x(b) = 0$. Therefore, we have the following updated feasible solution.



The corresponding residual graph is



We know that a flow, x^* is a maximum flow if and only if the residual network $G(x^*)$ contains no augmenting path. However, there does exist a possible augmenting path from $s, a, ...$ Therefore, we cannot conclude that the flow found is optimal.

4

Suppose that there are n people located throughout a maze, and there are n exits from the maze. You can view the maze as a graph $G = (V, A)$ where each edge (i, j) corresponds to a stretch of the maze where one can only continue "straight ahead" in that direction (from i to j) without any option to divert on a different path. While most edges come in pairs (i, j) and (j, i), reflecting the fact that if you go from i to j, it is also possible to go from j to i, there are also some passageways (such as slides) that may only be used in one direction (and hence only one of the two edges is in the graph). Each person is currently located at a node in the graph, and each exit is located at a node. Each exit allows only one person to go through; after the first person, the door locks and nobody else can pass.

a.) *Show that you can solve the problem of deciding if all n people can get out of the maze by solving an input to the maximum flow problem. You should explain the following*

1. How to set up an input to the maximum flow problem.
2. How to interpret the solution to the maximum flow problem as an answer to the maze problem.
3. Why your formulation is correct: so you need to explain
   - if the the maximum flow tells you (according to your interpretation of step 2) that there is a way to get all n people of the maze, why this is indeed the case.
   - if the the maximum flow tells you (according to your interpretation of step 2) that there is a *no* way to get all n people of the maze, why this is indeed the case.

Note: if you rely on the integrality property for your solution, you must explicitly explain where you use it and how it is relevant.

1.) Note that we can formulate this maze problem mathematically where $x_{i,j}$ is the number of people travelling along arc $x_{i,j}$.

$$\begin{aligned} &\max \sum_{s,j} x_{s,j\in A} \\ &\text{s.t. } \sum_{i,j\in A} x_{i,j} = \sum_{j,i\in A} x_{j,i} \forall i \in N/\{s,t\} \\ &\quad 0 \leq x_{i,j} \leq 1 \forall i,j \in E \end{aligned}$$

When seeking to formulate the maze problem as a maximum flow problem, we must begin by adding source and sink nodes s and t along with the corresponding arcs to connect people nodes to the source node and exit nodes to the sink node. Thus, the new graph can be denoted as $G' = (V', A')$ where $V' = V \cup \{s,t\}$ and $A' = A \cup \{(s,i) : i \text{ is a Person Node}\} \cup \{(j,t) : j \text{ is an Exit Node}\}$. Since our maze has the restriction that only one person may go through any exit, we will set all arc capacities $u_{ij} = 1$.

Now, we must prove that any feasible solution x to the original maze problem corresponds to a feasible solution x' to the max-flow problem. We know that x' has more elements than x since we added arcs in the construction of G'. However, when constructing our solution x' we can ensure that $x'_{i,j} = x_{ij} \forall (i,j) \in V$ so that the original maze problem's structure is maintained in the max-flow formulation. The additional elements represents the arcs connecting the people nodes to the source and the exit nodes to the sink, which we can represent in the solution x' by letting $x'_{si} = \sum_{(i,j)\in V'} \forall i : i$ is a People Node and $x'_{jt} = \sum_{(i,j)\in V'} \forall j : j$ is an Exit Node. Therefore, the solution to the max-flow problem, x', satisfies the conservation of flow constraints for both the People and Exit nodes since $\sum_{i,j\in A} x'_{ij} = \sum_{j,i\in A} x'_{ji}, \forall i \in V$.

Now, we must show that any feasible x' for the max-flow satisfies all of the constraint of the maze problem. We know that since the net flow coming in at any node is constrained by the capacity constraint of one, then the flow coming out at any node must also be one by the conservation of flow constraint. Thus, the net flow out of each People node can be at most one and the net flow out of each Exit node can be at most one, which satisfies the constraints of the maze problem.

Finally, we must demonstrate that the optimal solution to the max-flow problem that we have formulated is optimal for the original maze problem. The Integrality property tells us that our optimal solution will be integer since the input capacity values for the max-flow problem are integer. This is a necessary condition for the maze problem, since we can't have allow non-integer portions of people running about the maze. Finally, since the objective function for both problems is the same (maximize x_{ij}), the optimal values for the problems will be the same. Thus, we know that the optimal solution for the max-flow problem is the optimal solution for the maze problem.

b.) *Unfortunately that was not the right problem. Once someone uses the passageway (i, j) it is no longer possible to traverse the passageway from j to i (even if it was initially feasible). Explain (using the steps as in part (a)) how to solve this new problem as a maximum flow problem.*

We can begin with the same initial formulation for the maze problem as used above where $x_{i,j}$ is the number of people travelling along arc $x_{i,j}$.

$$\max \sum_{s,j} x_{s,j \in A}$$
$$\text{s.t.} \sum_{i,j \in A} x_{i,j} = \sum_{j,i \in A} x_{j,i} \forall i \in N/\{s,t\}$$
$$0 \leq x_{i,j} \leq 1 \forall i, j \in E$$

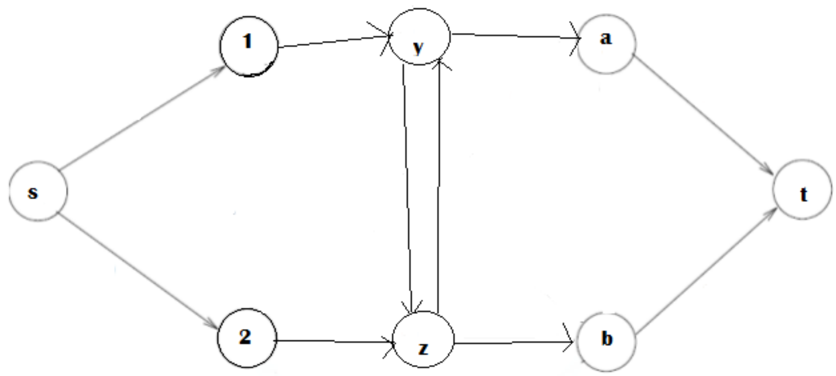
Also, as before, we can construct a feasible solution x' to the max-flow problem which corresponds to a feasible solution x to the original maze problem. We know that x' has more elements than x since we added arcs in the construction of G'. However, when constructing our solution x' we can ensure that $x'_{i,j} = x_{ij} \forall (i, j) \in V$ so that the original maze problem's structure is maintained in the max-flow formulation. The additional elements represents the arcs connecting the people nodes to the source and the exit nodes to the sink, which we can represent in the solution x' by letting $x'_{si} = \sum_{(i,j) \in V'} \forall i : i$ is a People Node and $x'_{jt} = \sum_{(i,j) \in V'} \forall j : j$ is an Exit Node. Therefore, the solution to the max-flow problem, x', satisfies the conservation of flow constraints for both the People and Exit nodes since $\sum_{i,j \in A} x'_{ij} = \sum_{j,i \in A} x'_{ji}, \forall i \in V$.

The main difference is that with the additional constraint in the maze problem, the corresponding solution must enforce a no-cycling constraint. This is because if a cycle occurs in the max-flow problem, then the corresponding situation in the maze problem is one in which a path is travelled twice. In the case that a cycle would appear in the solution to a max-flow problem, we would reduce the flow on both arcs involved in the cycle from 1 to 0. This would redirect the flow (the people) to other nodes, removing the cycle and satisfying the constraint that a path is useable only once.

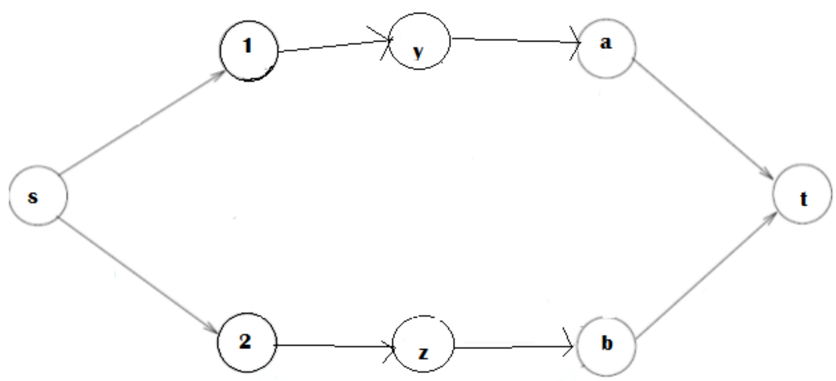
As before, we must demonstrate that the optimal solution to the max-flow problem that we have formulated is optimal for the original maze problem. The Integrality property tells us that our optimal solution will be integer since the input capacity values for the max-flow problem are integer. This is a necessary condition for the maze problem, since we can't have allow non-integer portions of people running about the maze. Finally, since the objective function for both problems is the same (maximize x_{ij}), the optimal values for the problems will be the same. Thus, we know that the optimal solution for the max-flow problem is the optimal solution for the maze problem.

c.) *Unfortunately that was still not the right problem. Each has a key, and each exit has a locked door (for which exactly one person has the key). Explain why the approach that described in the previous part does not work by giving an example in which your approach has the wrong answer (i.e, either your procedure in (b) says that all people cannot exit, when this is in fact possible, or vice versa).*

In a situation where each person has a key to a specific locked exit door, if we were to use the approach in part **b.)** an incorrect answer would be obtained since part of the formulation involved rerouting people to avoid cycling. Thus, people would be redirected to exit doors for which they do not have a key and they would be unable to leave the maze. For example, see the figure below.



We see that Person 1 with a key for exit door b takes the path from node y to node z on its route out of the maze. Similarly, Person 2 has a key for the exit door a takes the path from node y to node z on its route. This type of cycling violates the constraint formulated in part **b.)**. In accordance with that formulation, we will decrease the flow on the paths (y, z) and (z, y) to 0 and redirect the people. This will result in the following flow graph.



We see, however, that this results in person 1 attempting to exit from door a and person 2 attempting to exit from door b. However, they do not have the corresponding keys, which means they can't exit from these doors. Therefore, the feasbile solution for the max flow problem doesn't translate to a feasible solution to the maze problem with the added constraint.