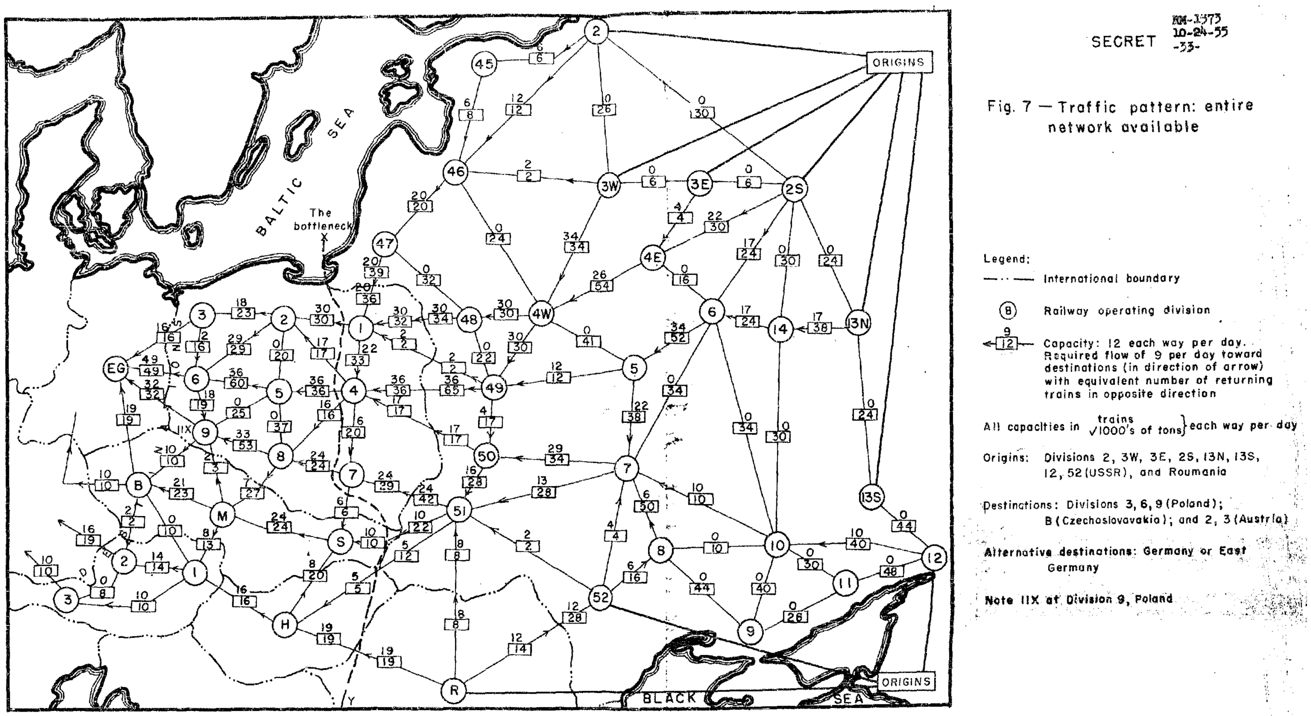
CS 225

Data Structures

December 6 – Maximum Flow

G Carl Evans

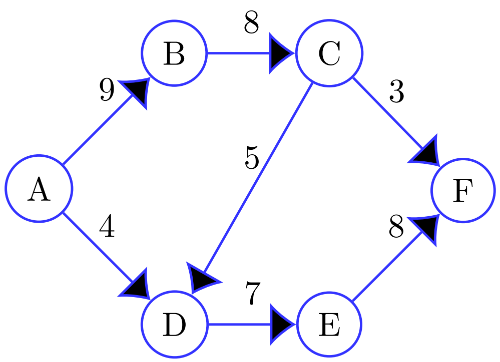
Origin of Maxflow Problem



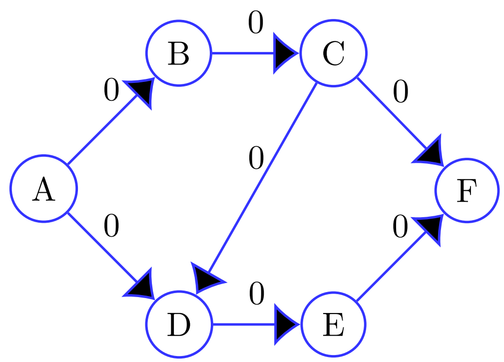
We are given as input graph $G1$.

We create two new graphs: a *flow graph* F and a *residual graph* R.

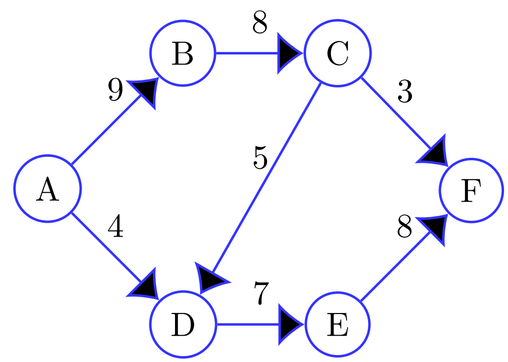
Graph G



Flow Graph F



Residual Graph R

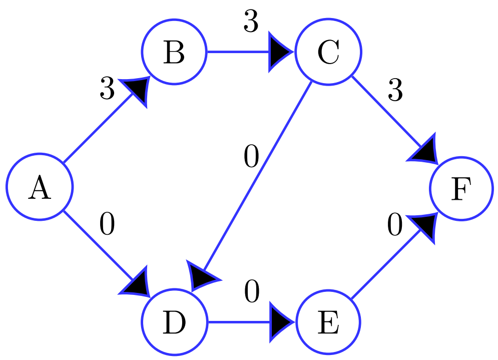


Problem 1.

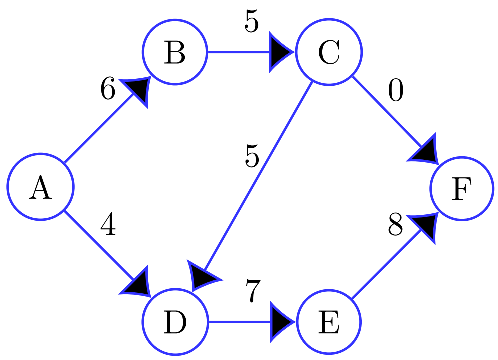
The algorithm works by selecting paths from the residual graph R. The first path selected is $A \to B \to C \to F$ in graph R. This path's flow capacity is 3. What do you think determines the flow capacity?

The algorithm uses the path to modify graphs F and R. Here is the result.

Graph F



Graph R



Problem 2.

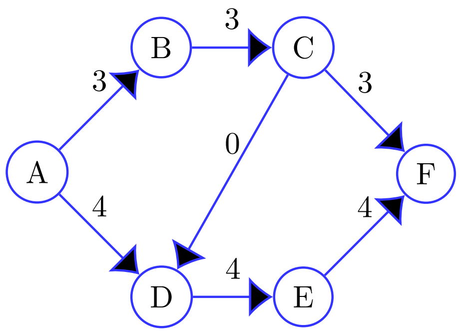
Examine the new versions of F and R above. What is being done with the path selected from R to modify these graphs?

Problem 3.

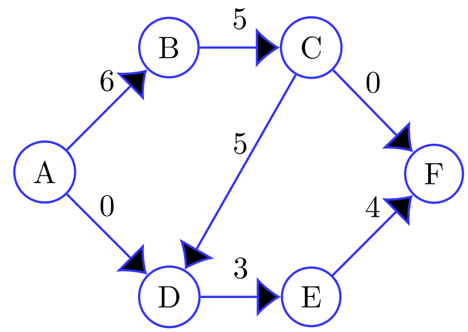
The next path selected was $A \to D \to E \to F$ in graph R. What is the flow capacity of that path?

The resulting working graphs are these:

Graph F



Graph R



Problem 4.

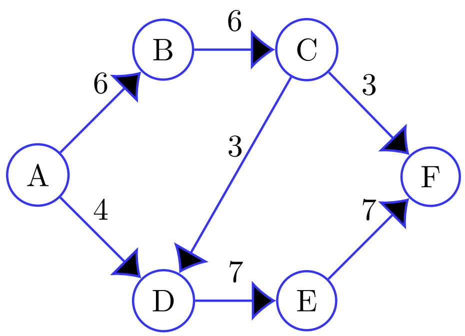
We select path $A \to B \to C \to D \to E \to F$. What is the flow capacity of that path?

Problem 5.

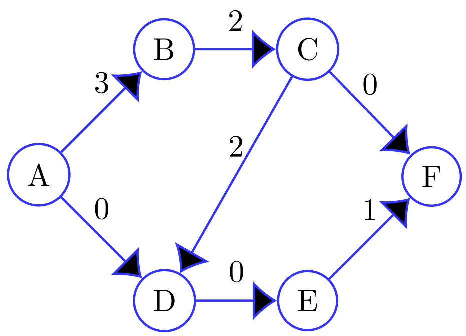
The paths selected always start from node A and end with node F. What is different about these nodes compared to the others?

Here are the final working graphs F and R.

Graph F



Graph R



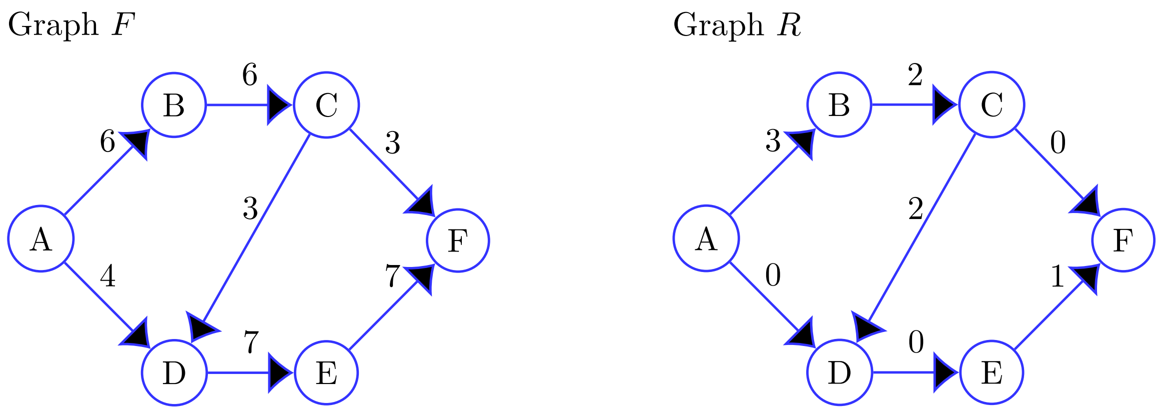
Problem 6.

At this point, the algorithm is finished. How can we know the algorithm is done by examining graph R?

Problem 7.

For nodes B, C, D, and E, what is the relationship between the in-flows and the out-flows? Why does that relationship have to exist?

Here are the final working graphs F and R.



Problem 8.

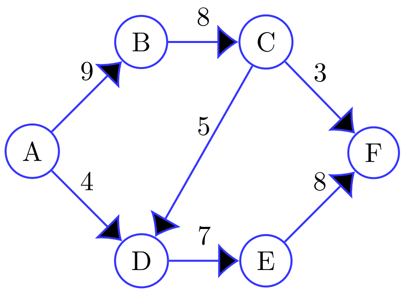
Using the final flow graph F above, determine the maximum flow of graph $G1$.

Problem 9.

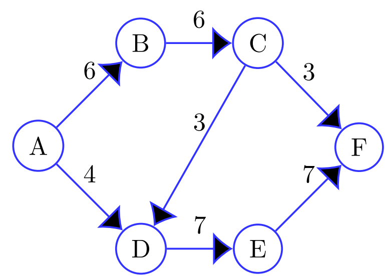
In graph F, the outflow of A is equal to the inflow of F. Should that always be the case?

Here are the final working graphs F and R.

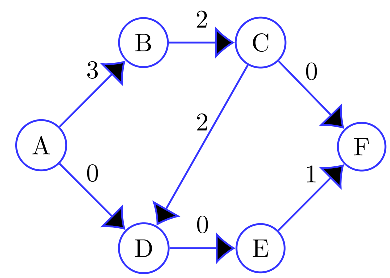
Graph G



Graph F



Graph R

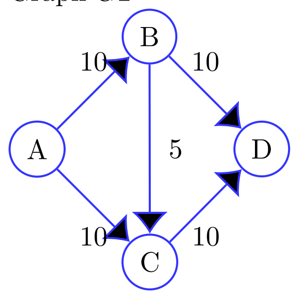


Problem 10.

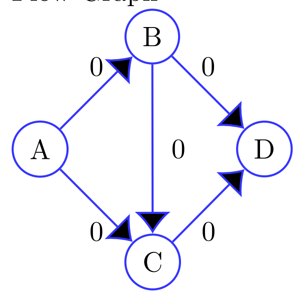
Node A is called a *source node* and node F is called a *sink node*. Would this technique work if there were multiple source and sink nodes? Why or why not?

Now we are going to look at a case that messes up the algorithm.

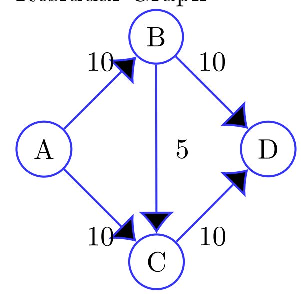
Graph $G2$



Flow Graph



Residual Graph



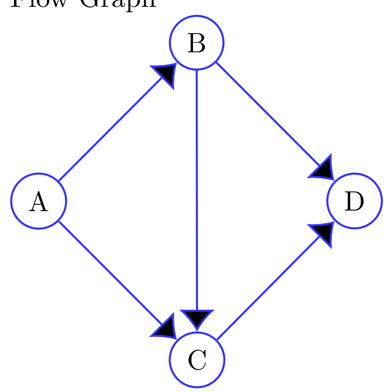
Problem 11.

The algorithm picks path $A \to B \to C \to D$. What is the capacity of that path?

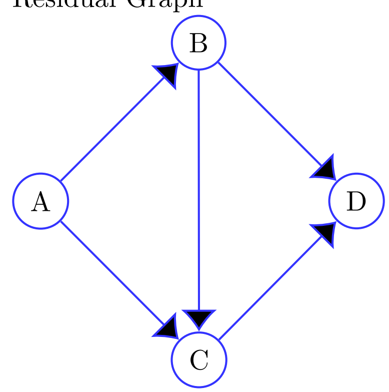
Problem 12.

Update the flow and residual graphs as a result of selecting this path.

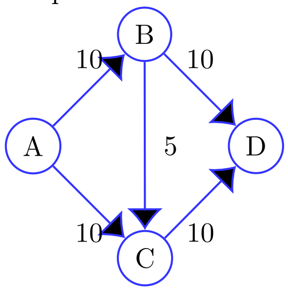
Flow Graph



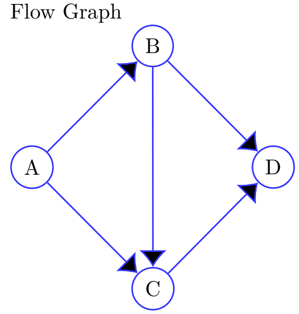
Residual Graph



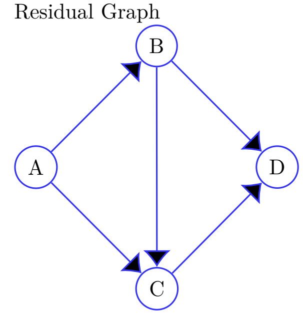
Graph $G2$



Flow Graph



Residual Graph



Problem 17.

What is the maximum network flow of $G2$, according to the algorithm?

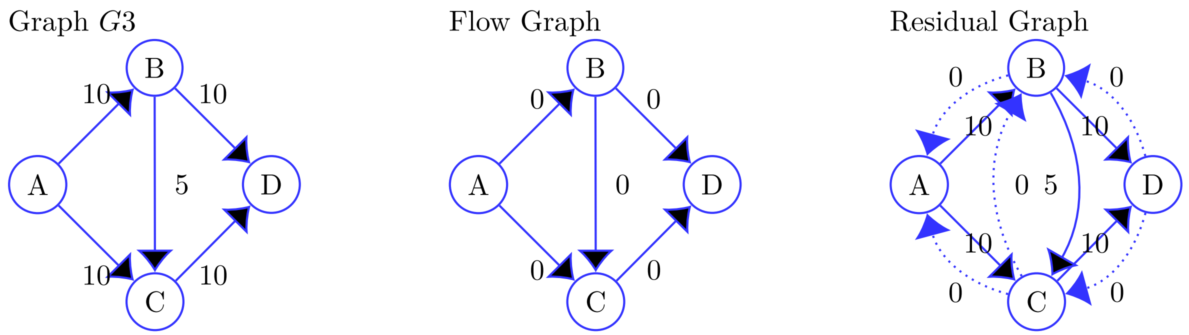
Problem 18.

Is this number correct? Why or why not? Examine $G2$ to verify your answer.

Problem 19.

Suppose $G2$ modeled a network of water pipes. What would happen on edge $B \to C$ in this situation? Would it change the total flow of $G2$ if we deleted that edge?

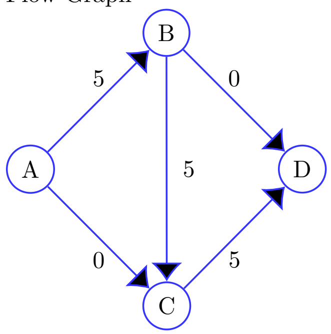
We are going to modify the algorithm. Starting again with the previous graph, we make a new kind of residual graph. The dotted edges are added, and are legal edges to be traversed in the residual graph.



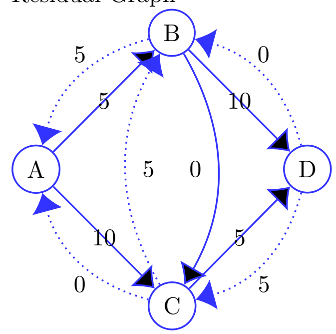
Problem 20.

Select path $A \to B \to C \to D$. What is the capacity of that path?

Flow Graph

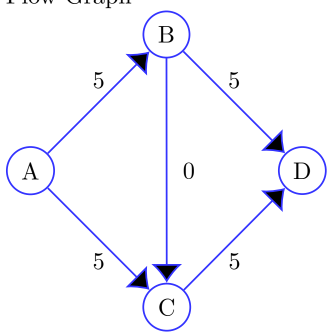


Residual Graph

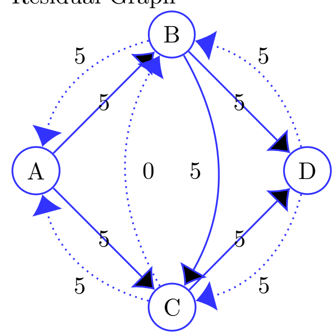


Now we select path $A \to C \to B \to D$.
Here are the updated flow and residual graphs:

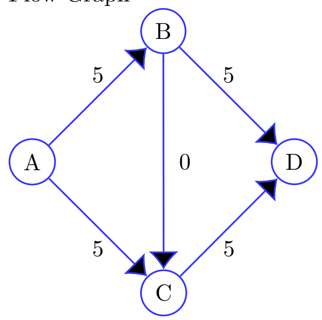
Flow Graph



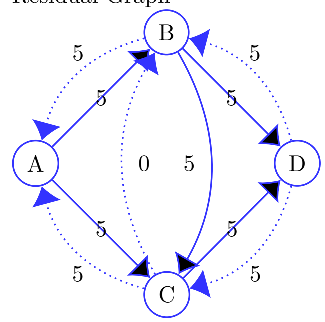
Residual Graph



Flow Graph



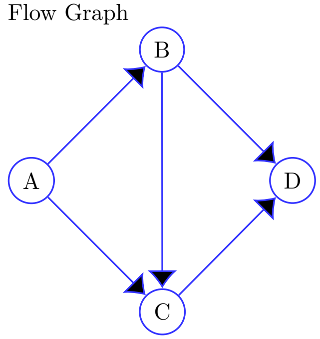
Residual Graph



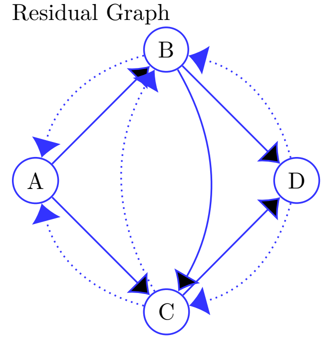
Problem 22.

Select path $A \to B \to C \to D$. (Yes, we are repeating this path.) What are the resulting flow and residual graphs?

Flow Graph



Residual Graph

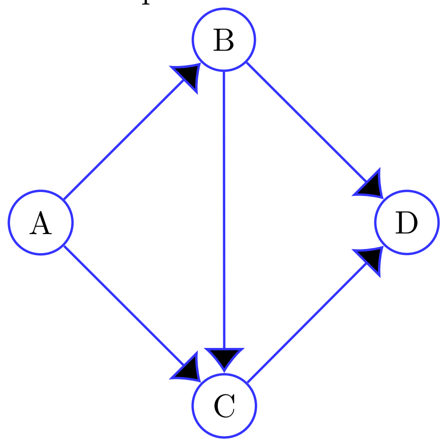


Problem 23.

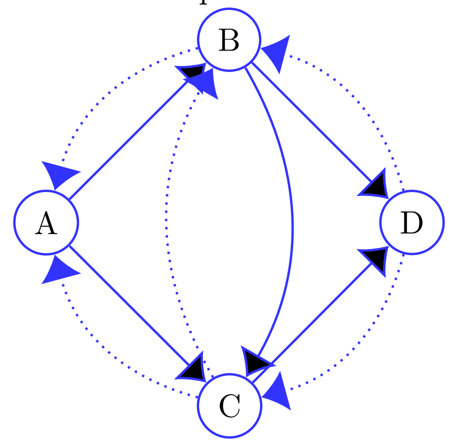
Now we select path $A \to C \to B \to D$.

What are the updated flow and residual graphs?

Flow Graph



Residual Graph



Problem 24.

At this point, the algorithm should be done. Is the final network flow accurate now?

Ford Fulkerson Requirements and Runtime

Ford Fulkerson Problems

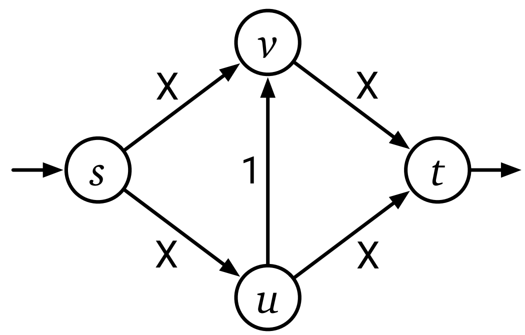


Figure 10.7. Edmonds and Karp's bad example for the Ford-Fulkerson algorithm.

Image from https://jeffe.cs.illinois.edu/teaching/algorithms/book/10-maxflow.pdf

Edmonds and Karp’s Algorithms