Quiz 3 (Maximum Flow Problem)

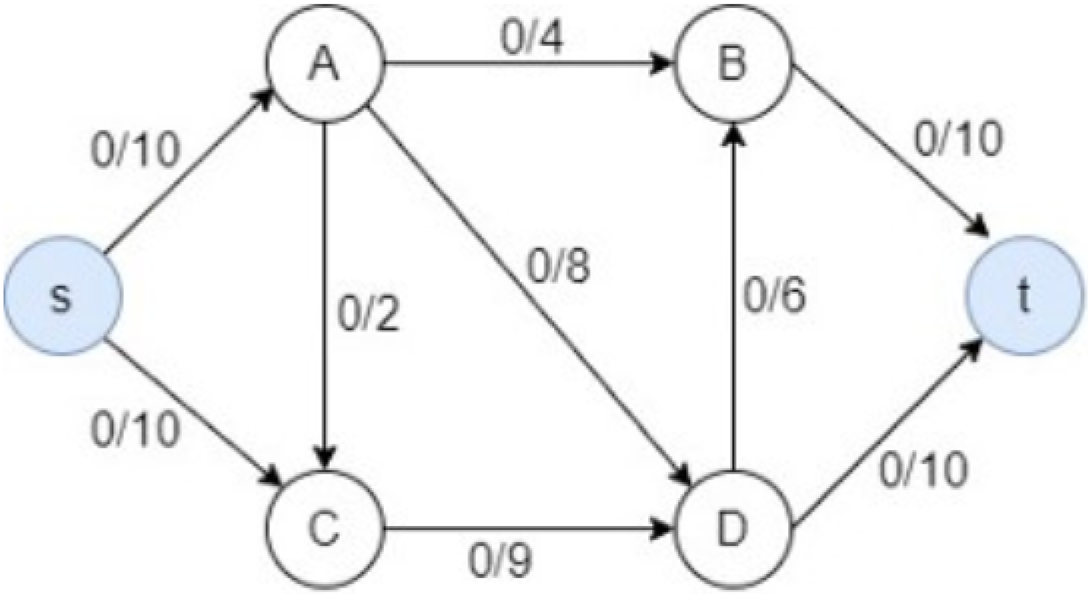
Quiz

1) A network can have only one source and one sink. 2)Ford Fulkerson's Algorithm in fact uses the idea of Naïve Greedy Algorithm.

- ○ 1) is True ,2) is True.
- ○ 1) is True ,2) is False.
- ◉ 1) is False ,2) is True.
- ○ 1) is False,2) is False.

Clear selection

The maximum flow for the given graph is (where the first value is the flow value and the second value is the capacity of the edge)?



- (selected) Eighteen
- Nineteen
- Twenty
- Twenty one
- None of the above

Clear selection

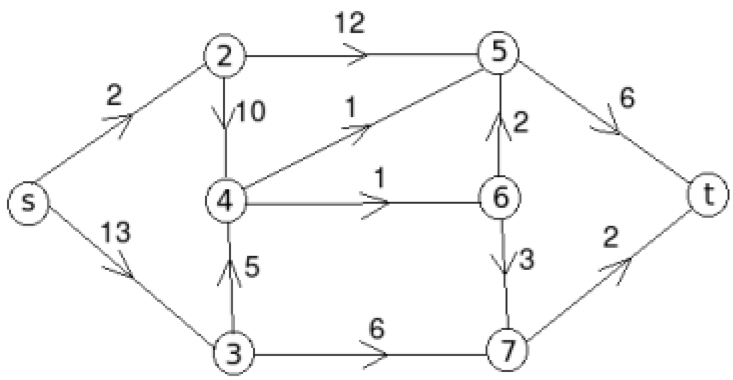
A given graph can have multiple minimum capacity cut. Let (A,B) be an s-t cut in a flow network G. The cut (A,B) may not be minimum capacity cut. fout(A) represents flow out of A and fin(A) represents flow into A. Which one holds true for the cut (A,B):

- fout(A) = fout(B)
- Fin(A) = Fin(B)
- Fout(A) =Fin(B)
- no such relation exists

You are given an undirected graph with each edge having a capacity of 1 unit (i.e., a maximum of one unit of water can flow in both directions). Suppose you ran the Ford-Fulkerson algorithm between a pair of vertices s and t on this graph and it terminated with a final flow of 1. Which of the following is the most general statement one can make about the original graph?

- There exists only one unique path from s to t in the graph.
- There exists one edge which can be removed to disconnect the graph into two pieces, one containing s and the other containing t.
- The graph is a tree, with s as the root and t as one of the leaves.
- None of the above, as network flow only works for directed graphs

Find the maximum flow in the following graph.



- Ten
- Five
- Seven
- Six

The first step in the naïve greedy algorithm is?

- adding flows with higher values
- reversing flow if required
- analyzing the zero flow
- calculating the maximum flow using trial and error

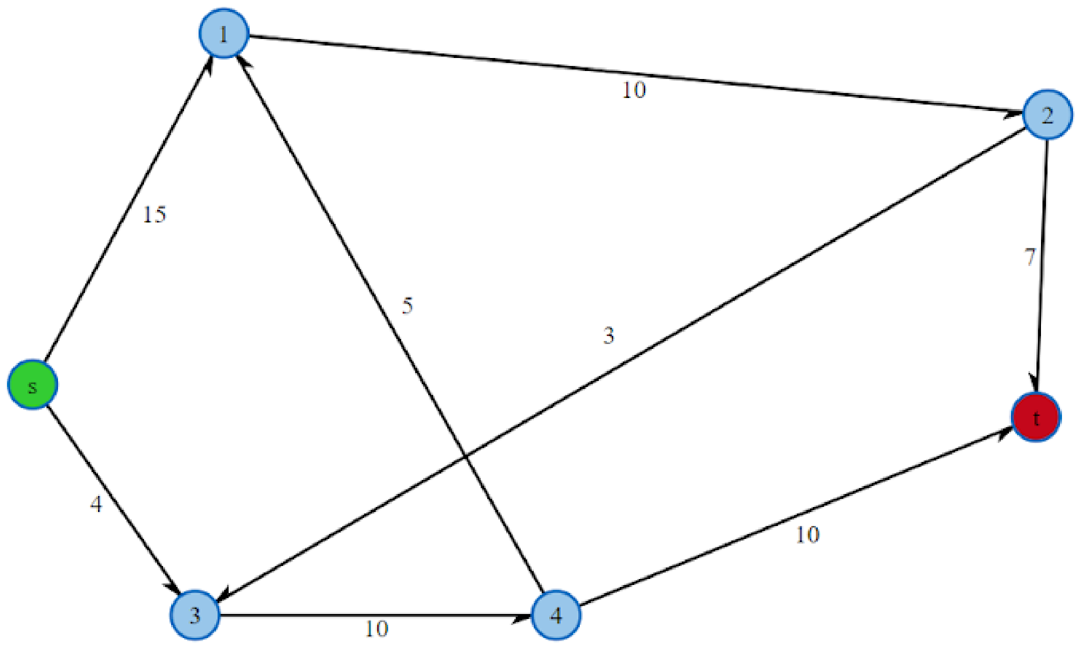
An efficient max-flow algorithm can be used to efficiently compute a maximum matching of a given bipartite graph.

- F(flase)
- T(true)
- Can't be determined
- may be false

Consider a flow network (graph) G(V,E) and some cut(A, B) such that s belongs to A and t belongs to B and some flow value v(f) corresponding to some feasible flow f(follows capacity and conservation conditions). Which of the following holds true always

- v(f) ≤ capacity of cut(A,B)
- max flow value > minimum capacity of any s-t cut
- max flow value < minimum capacity of any s-t cut
- v(f) = capacity of cut(A,B)

Find the Maximum Flow for the given Graph.



- Nineteen
- Fifteen
- Seventeen
- Fourteen

Which algorithm is used to solve a maximum flow problem?

- Prim's Algorithm
- Kruskal's Algorithm
- Dijkstra's Algorithm
- Ford Fulkerson Algorithm

Initially, the amount of flow carried by all the edges are

- Variable in nature
- Constant in nature
- positive rational values
- positive integer values
- None of the above

Maximum possible number of iteration required for the termination of while loop from the Ford-Fulkerson Algorithm is

$$C_e$$

- Option 1

$$\sum_{e\ out\ of\ s} C_e$$

- Option 2

$$\sum_{e\ out\ of\ all\ v\ except\ t} C_e$$

- Option 3

$$\sum_{e\ in\ all\ v\ except\ s} C_e$$

- Option 4

What is the runtime complexity of Ford Fulkerson algorithm for finding out maximum flow in a directed graph G where all edges are integers? Assume that number of edges is m, number of vertices is n and it terminates in at most C iterations.

- O(m*n*C)
- O(n*C)
- O(m*C)
- O(m+n)

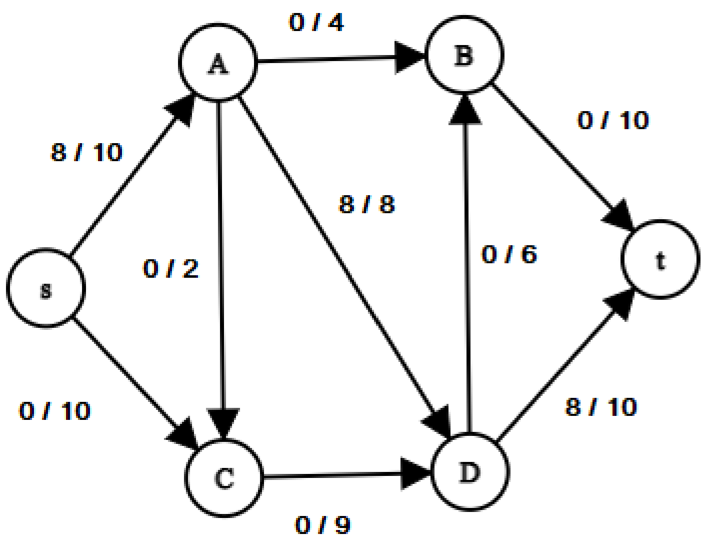
Let 'f' be any s-t flow ,and (A-B) any s-t cut . Then v(f) =

- fout(A) - fin(B)
- fout(B) - fin(B)
- fout(A) - fin(A)
- fout(B) - fin(A)

For any flow network G and any maximum flow on G, there is always an edge e such that increasing the capacity of e increases the maximum flow of the network.

- Depends on the given graph.
- T(true)
- F(flase)
- Can't be determined.

Consider the graph where the first value is the flow value and the second value is the capacity of the edge. What is the bottleneck capacity of the augmenting path s->C->D->t?



- Zero
- Two
- Eight
- Nine
- Ten

The time complexity of the Ford-Fulkerson algorithm for computing maximum flow in a graph where edge weights are positive real numbers (not necessarily integers) is:(m is the number of edges and C is upper bound on number of iterations of while loop)

- O(m*C)
- O(C)
- May be exponential in m and C
- Ford-Fulkerson algorithm is not applicable to non-integral edge weights.

Suppose that you run the Ford-Fulkerson algorithm (using the shortest augmenting path heuristic) to solve a bipartite matching problem with N students and N companies. How many augmenting paths are needed in the worst case?

- N
- N^2
- 0.5 N^3
- N^3

What is the objective of a maximum flow problem?

- Maximize the amount flowing through a network
- Maximize the profit of the network
- Maximize the routes being used
- None of the above

While augmenting path in a residual graph, If e is the backward edge then (where b is bottleneck and Ce is edge capacity)

- Increase f(e) in G by Ce
- Decrease f(e) in G by Ce
- Increase f(e) in G by b
- Decrease f(e) in G by b

G be given graph , Gf be residual graph of G , then

- Gf has at most as many edges as G.
- Gf has at most twice as many edges as G.
- G has at most twice as many edges as Gf.
- Gf has atleast twice as many edges as G.

Given a flow network, let f be any flow and let (A,B) be any cut. Then, the net flow across (A,B) is __________ the value of f.

- less than
- greater than
- equal to
- not related to

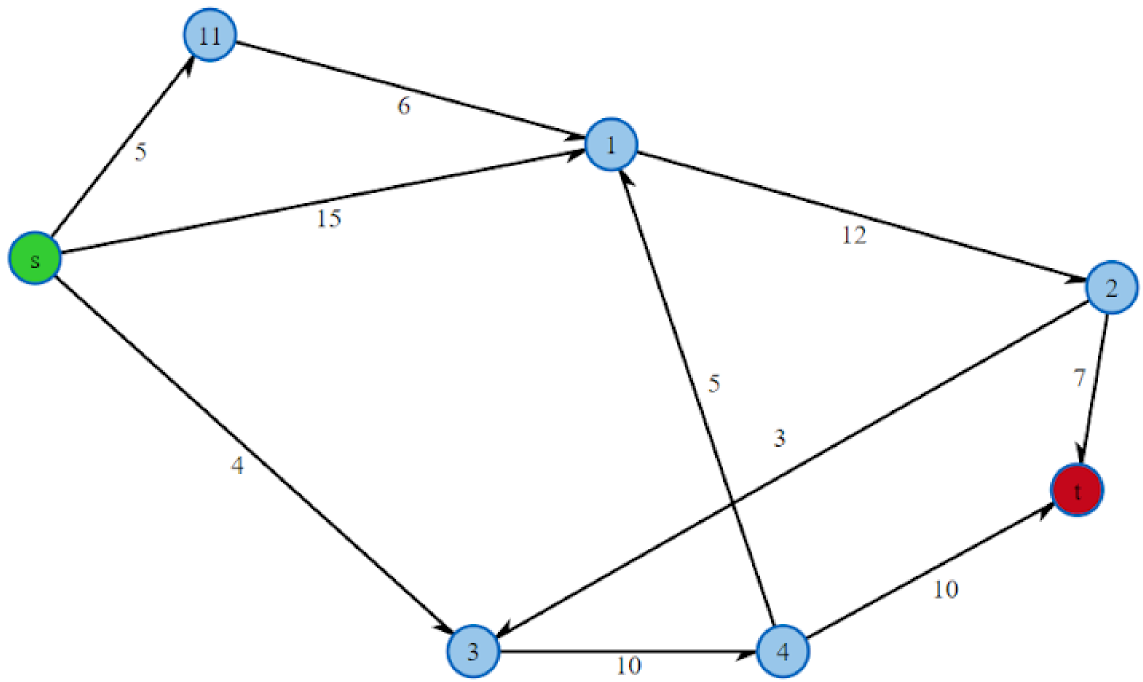
An augmented path can be found in

- Linear time
- Quadratic time
- Exponential time
- Logarithmic time

What is the running time of an unweighted shortest path algorithm whose augmenting path is the path with the least number of edges?

- O(|E| log |V|)
- O(|E|)
- O(|E|^2 |V|)
- O(|E|^2 log |V|)

Find the maximum flow for the following graph.



- Fourteen
- Eighteen
- Fifteen
- Twenty two

What is the source in the maximum flow problem?

- (selected) Vertex with no incoming edges
- Vertex with no leaving edges
- Centre vertex
- Vertex with the least weight

Clear selection

Which of the following is true?

- On augmentation, flow value increases then decrease
- On augmentation, flow value decreases then increase
- On augmentation, flow value strictly decreases
- On augmentation, flow value strictly increases
- The nature of the flow value can not be determined.

If e=(u,v) is a forward edge then its residual capacity is

- Ce - f(e)
- f(e) – Ce
- Ce+f(e)
- Ce

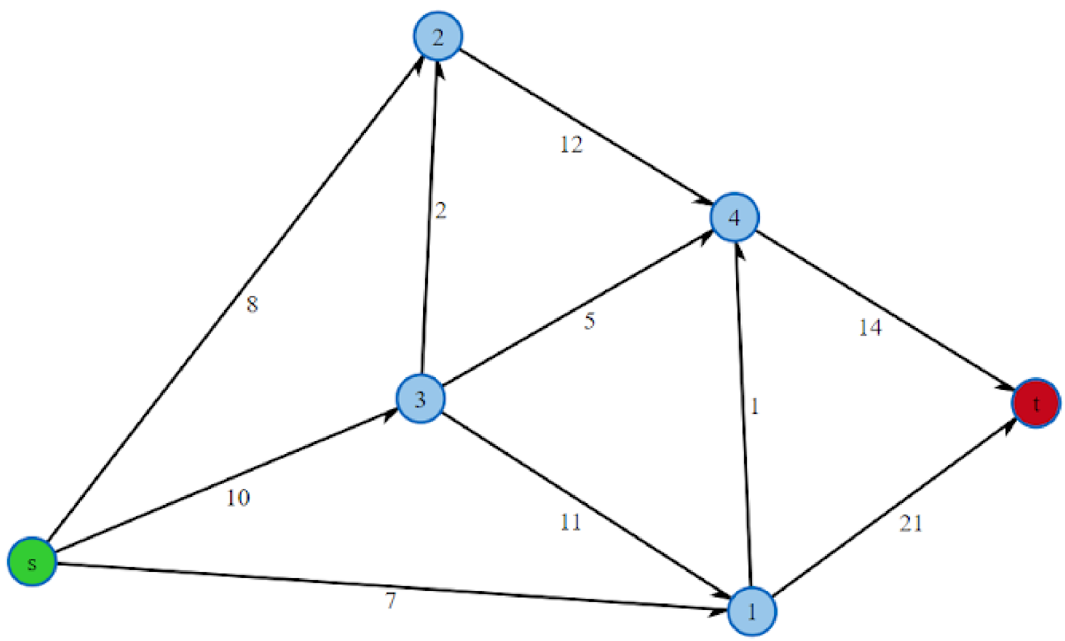
Given an unweighted directed flow network G(V,E), the goal is to find the number of edge disjoint paths from s to t. This can be done by assign a weight of 'x' to each of the edge e in E and executing Ford-Fulkerson algorithm on the weighted graph G. The smallest value of 'x' can be:

- Zero
- One
- square root of n
- square root of m

How many constraints are there in the Maximum-Flow problem?

- One
- Two
- Three
- Four

Find the Maximum Flow for given graph.



- Twenty
- Twenty Three
- Twenty Five
- Eighteen

A given graph G(V,E) has a unique minimum cut. The number of edges in minimum cut(size of minimum cut) is k. The cardinality of set E is m and cardinality of set V is n. The number of edges in the final residual graph after the Ford-Fulkerson algorithm is finished is:

- at most m-n
- exactly 2*m-n
- at most 2*m-k
- exactly 2*m-k

A simple acyclic path between source and sink which passes through only positive weighted edges of the residual graph is called?

- Augmenting path
- Critical path
- Residual path
- Maximumpath

Let 'C' be the upper bound on flow that can come out of 's' in a flow graph G. All capacities in the flow network G are multiple of log (n). Base of log is 2. The best upper bound on the time complexity of Ford- Fulkerson algorithm to find maximum flow in this flow network G is

- O(m*C)
- O(m*C*log (n))
- O(m*C/log(n))
- O(m*log(C))

Under what condition can a vertex combine and distribute flow in any manner?

- It should maintain flow conservation
- It may violate edge capacities
- The vertex should be a source vertex
- The vertex should be a sink vertex

Let v be the number of nodes, e be number of edges and U be the capacity of the largest edge of a graph G. What is the time complexity when Ford Fulkerson algorithm is applied to find the maximum flow?

- O(v*e^2)
- O(v^3)
- O(v^2*e*loge)
- O(v*e*U)

Suppose you have a flow network G with integer capacities, and an integer maximum flow f. Suppose that, for some edge e, we increase the capacity of e by one.

- An O(E) time algorithm is possible to find a maximum flow in the modified G.
- An O(E) time algorithm is not possible to find a maximum flow in the modified G.
- An O(V+E) time algorithm is possible to find a maximum flow in the modified G.
- None of these.

Consider the flow graph after few iterations of while loop of Ford-Fulkerson algorithm (each iteration finished completely) and the flow value is less than maximum possible flow. Then, the corresponding flow graph:

- ○ Does not follow capacity condition
- ○ Does not follow conservation condition
- ○ represents a feasible flow
- ○ implies non-existance of augmenting path

How many edges does the residual graph have in the worst-case scenario? Where n is the number of nodes in G and m is the number of edges in G.

- ○ m + m*n
- ○ m*n
- ○ m+m-2
- ◉ m+m
- ○ m - 2

Clear selection

Ford-Fulkerson algorithm uses

- ○ Dynamic Programming
- ◉ Greedy approach
- ○ Divide and Conquer
- ○ Binary Search

Clear selection

Augmenting Path can be found in

- ◯ O(|E | log |V|)
- ◉ O(|E|)
- ◯ O(|E^2|)
- ◯ O(|E|^2 log |V|)

Clear selection

Page 3 of 3

Back Submit

Never submit passwords through Google Forms.