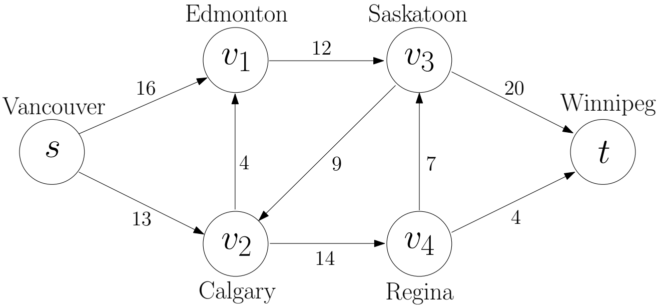
COMP3711: Design and Analysis of Algorithms

Tutorial 11

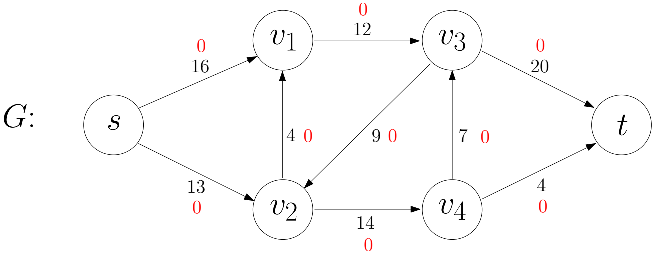
HKUST

Question 1

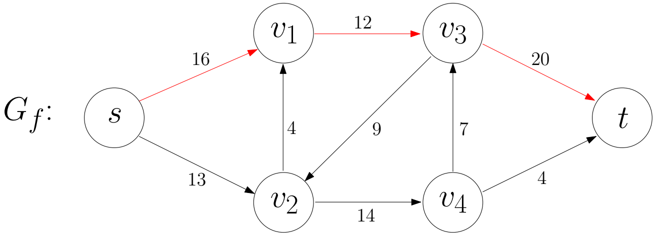
Show the execution of the Edmonds-Karp algorithm on the following flow network.



Solution 1

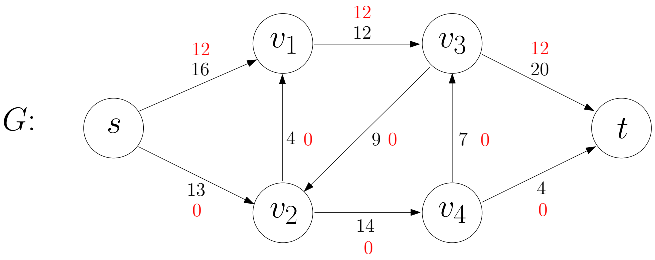


G:
s
v1
v2
v3
v4
t
0
16
0
12
0
20
4 0
9 0
7 0
13
0
14
0
4
0

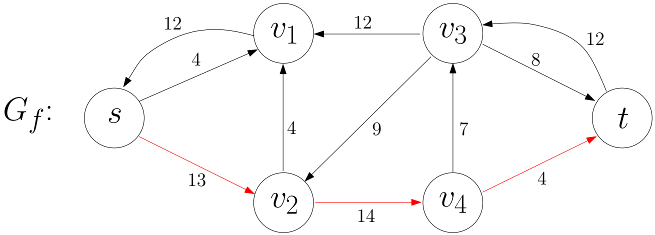


Gf:
s
v1
v2
v3
v4
t
16
12
20
4
9
7
13
14
4

Solution 1

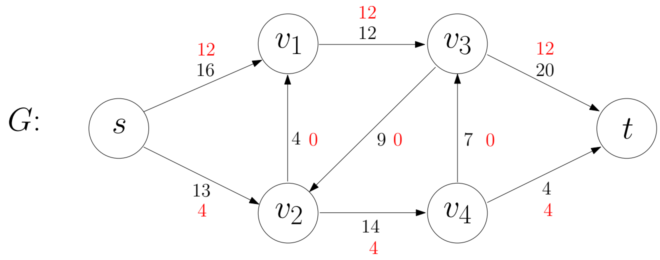


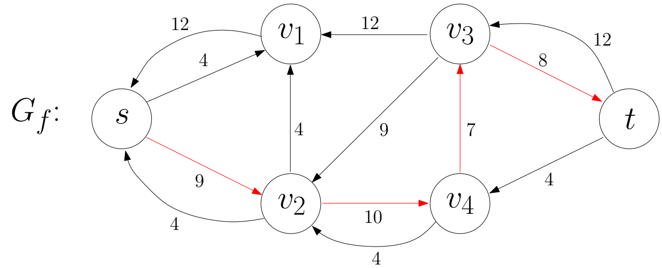
G:
s
v1
v2
v3
v4
t
12
16
12
12
12
20
4 0
9 0
7 0
13
0
14
0
4
0



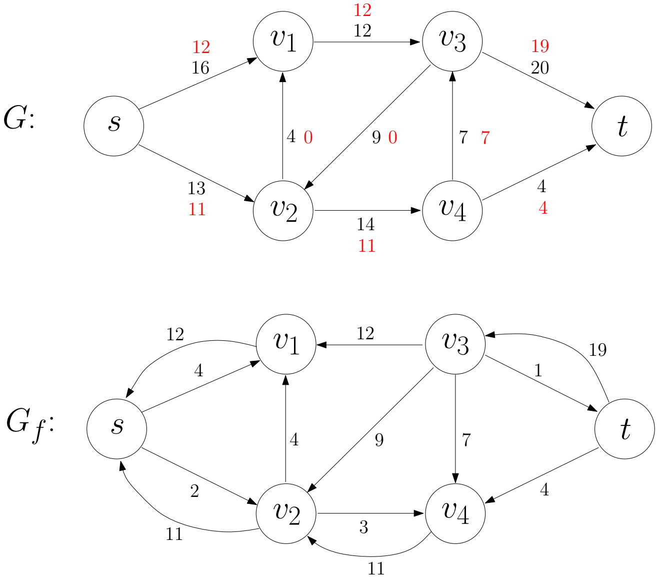
Gf:
s
v1
v2
v3
v4
t
12
4
12
12
8
4
9
7
13
14
4

Solution 1

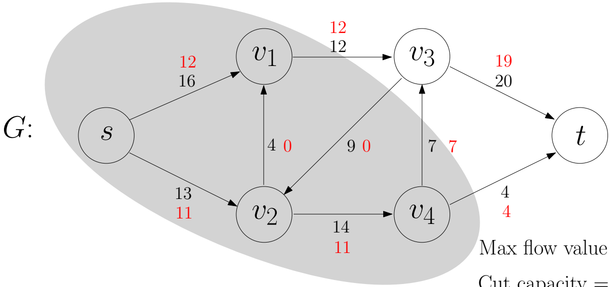

G:
s
v1
v2
v3
v4
t
12
16
13
4
12
12
4 0
9 0
7 0
12
20
14
4
4
4


Gf:
s
v1
v2
v3
v4
t
12
4
12
12
8
4
9
7
9
4
10
4
4

Solution 1

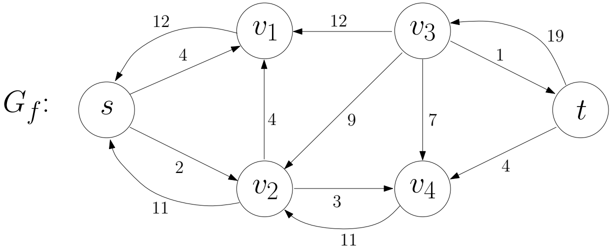

G:
s
v1
v2
v3
v4
t
12
16
13
11
12
12
4 0
9 0
7 7
19
20
4
4
14
11
Gf:
12
4
2
11
4
9
7
1
19
4
3
11
12

Solution 1



Max flow value = 23

Cut capacity = 23



Question 2

Extend the flow properties and definitions to the multiple-source, multiple-sink problem. Show that any flow in a multiple-source, multiple-sink flow network corresponds to a flow of identical value in the single-source, single-sink network obtained by adding a supersource and a supersink, and vice versa.

Solution 2

Suppose that there are multiple sources $s_1, s_2, \ldots, s_m$ and multiple sinks $t_1, t_2, \ldots, t_n$ of the flow network G.

The definition becomes to find a flow f with maximum value

$$|f| = \sum_{i=1}^{m} \sum_{v} f(s_i, v) = \sum_{i=1}^{n} \sum_{v} f(v, t_i).$$

The flow conservation property becomes

$$\forall v \in V - \{s_1, s_2, \ldots, s_m, t_1, t_2, \ldots, t_n\}, \quad \sum_{e \text{ out of } v} f(e) = \sum_{e \text{ into } v} f(e)$$

Solution 2

Obtain a flow network G' from G by adding a supersource s with edges (s, s_i) for $1 \leq i \leq m$ and a supersink t with edges (t_i, t), where the capacity of these edges are set to infinity. Then, any flow in G corresponds to a flow of identical value in G', and vice versa.

Question 3

The ***edge connectivity*** of an undirected graph is the minimum number k of edges that must be removed to disconnect the graph. For example, the edge connectivity of a tree is 1, and the edge connectivity of a cyclic chain of vertices is 2. Show how to determine the edge connectivity of an undirected graph $G = (V, E)$ by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(V)$ vertices and $O(E)$ edges.

Solution 3

Step 1: Choose a vertex $s \in V$ arbitrarily.

Step 2: For each vertex $v \in V - \{s\}$, build a flow network G_{sv} from G as follows,

- 2.1: Each undirected edge $(x, y) \in E$ becomes two directed edges (x, y) and (y, x) (auxiliary vertex can be added to avoid antiparallel edges)
- 2.2: Remove the incoming edge of s and outgoing edge of v.
- 2.3: Each directed edge has weight 1.

Step 3: For each flow network G_{sv}, obtain the maximum flow f_{sv} with source s and sink v.

Step 4: The edge connectivity $k = \min_{v \in V - \{s\}} |f_{sv}|$.

Solution 3

Correctness: For any cut $C = \{S, V - S\}$, if the edges in G that cross C are removed, then the graph is disconnected. Thus, the edge connectivity problem becomes finding the cut with the fewest edges crossing it. Without loss of generality, we assumes $s \in S$. The problem is that we don't know which vertex v must be in $V - S$, which is why we try all possible v.

For any flow network G_{sv}, the capacity of any cut C in G_{sv} is the same as the number of edges in G that cross C since each edge weight has 1. By the max-flow min-cut theorem, the capacity of the minimum cut of G_{sv} is the same as its maximum flow, so we can find the minimum number of edges needed to cut s and v apart. Finally, after trying all possible v, we can find the minimum number of edges needed to cut the graph into any two parts.