Practice Problems Set 12

Fall 25

1. Application of Maximum Flow (attributed to CLRS Exercise 24.1-6)

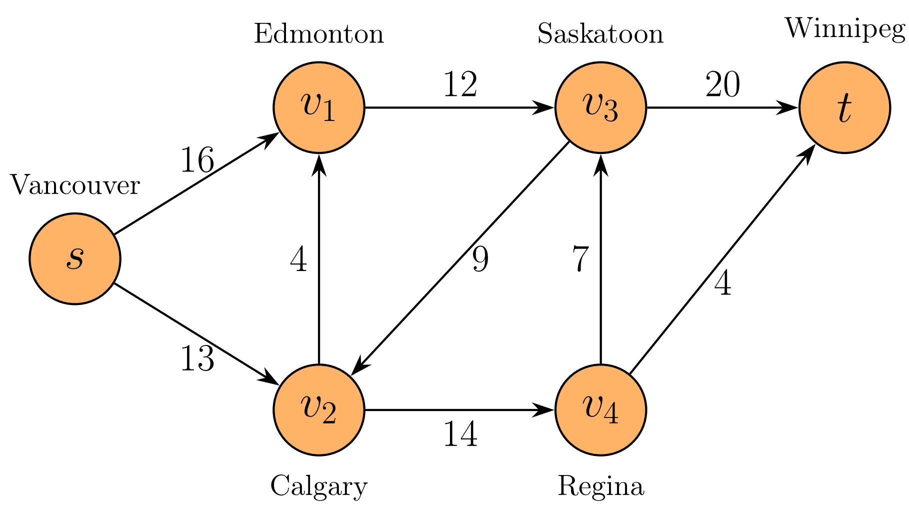
   Professor Adam has two children who, unfortunately, dislike each other. The problem is so severe that not only do they refuse to walk to school together, but in fact each one refuses to walk on any block that the other child has stepped on that day. The children have no problem with their paths crossing at a corner. Fortunately both the professor's house and the school are on corners, but beyond that he is not sure if it is going to be possible to send both of his children to the same school. The professor has a map of his town. Show how to formulate the problem of determining whether both his children can go to the same school as a maximum-flow problem.

Solution:

Create a vertex for each corner, and if there is a street between corners u and v, create directed edges u, v, u, v_u, and v_u, v, where v_u is a unique vertex created for only this street between corners u and v. (We need vertex v_u to avoid antiparallel edges. Note that if there is a street between corners u and v and between corners x and v, then the vertices v_u and v_x are distinct.) Set the capacity of each edge to 1. Let the source be the corner on which the professor's house sits, and let the sink be the corner on which the school is located. We wish to find a flow of value 2 that also has the property that $f(u, v)$ is an integer for all vertices u and v. Such a flow represents two edge-disjoint paths from the house to the school.

2. Edmonds-Karp Algorithm Practice (attributed to CLRS Exercise 24.2-3)

   Show the execution of the Edmonds-Karp algorithm on the running example flow network:

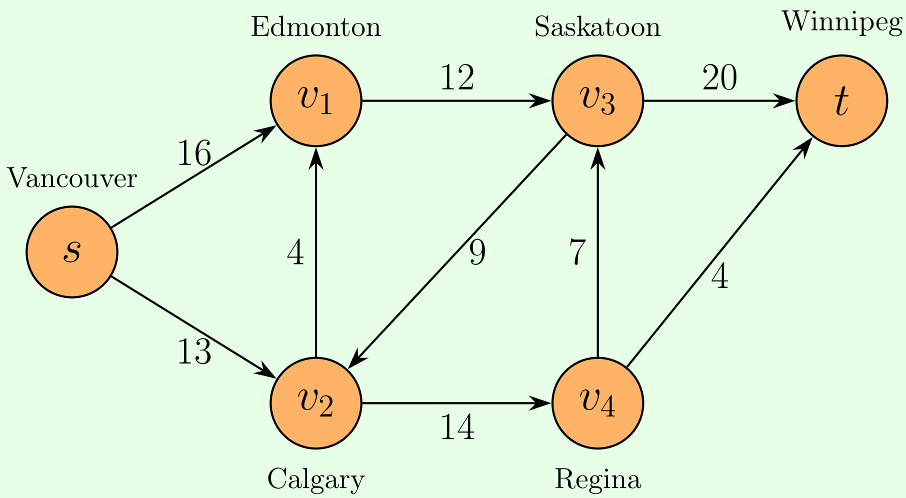


Solution:

The flow f is initially 0. The total flow $|f| = 0$.

Initial State (Iteration 0)

The initial flow network G is given. All flows are $f(u, v) = 0$. The residual graph G_f is identical to G.



Total Flow: $|f| = 0$.

Iteration 1

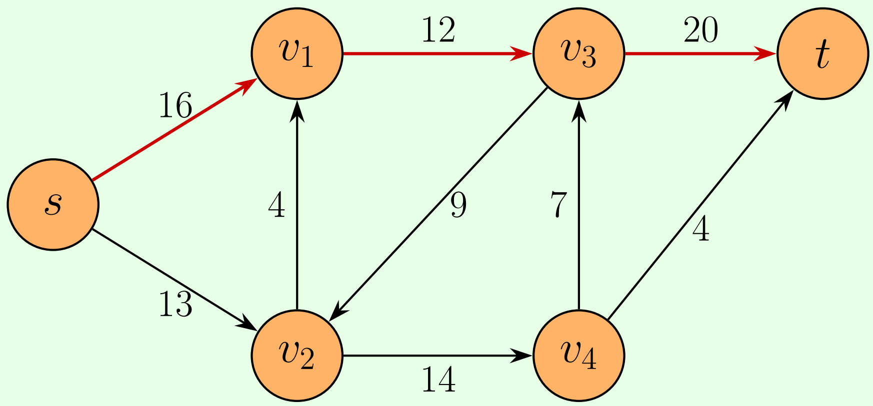
(a) **Find Path (BFS):** Run BFS on G_f. The shortest path is $p_1 = s \to v_1 \to v_3 \to t$ (length 3).

(b) **Bottleneck:** The residual capacity of p_1 is $c_f(p_1) = \min(c_f(s, v_1), c_f(v_1, v_3), c_f(v_3, t)) = \min(16, 12, 20) = 12$.

(c) **Augment Flow:** We add 12 units of flow along p_1.

(d) **Total Flow:** $|f| = 0 + 12 = 12$.

Residual Graph G_{f_1}: We update the residual capacities. The augmenting path is shown in red.



Iteration 2

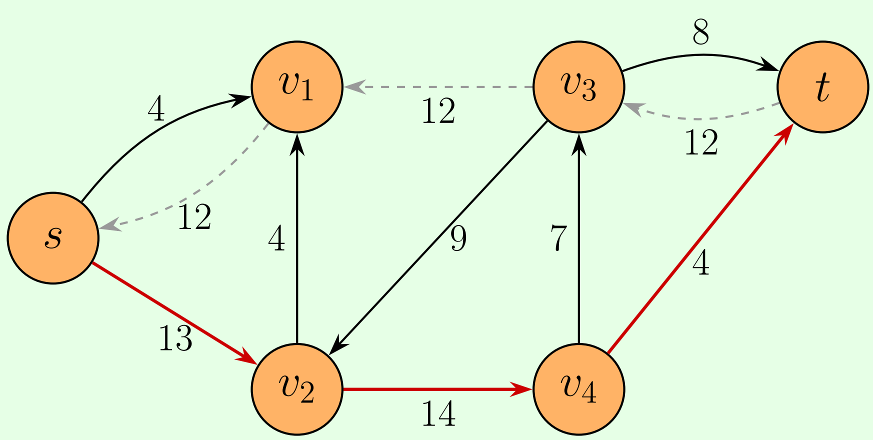
(a) **Find Path (BFS):** Run BFS on G_{f_1}. Edge (v_1, v_3) is saturated. The shortest path is $p_2 = s \to v_2 \to v_4 \to t$ (length 3).

(b) **Bottleneck:** $c_f(p_2) = \min(c_f(s, v_2), c_f(v_2, v_4), c_f(v_4, t)) = \min(13, 14, 4) = 4$.

(c) **Augment Flow:** We add 4 units of flow along p_2.

(d) **Total Flow:** $|f| = 12 + 4 = 16$.

Residual Graph G_{f_2}: The augmenting path p_2 is shown in red.



Iteration 3

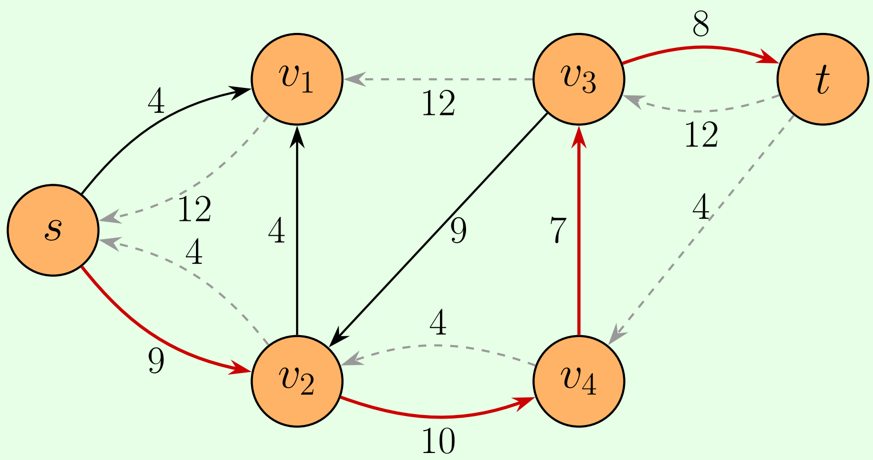
(a) **Find Path (BFS):** Run BFS on G_{f_2}. Edges (v_1, v_3) and (v_4, t) are saturated. The shortest path is $p_3 = s \to v_2 \to v_4 \to v_3 \to t$ (length 4).

(b) **Bottleneck:** $c_f(p_3) = \min(c_f(s, v_2), c_f(v_2, v_4), c_f(v_4, v_3), c_f(v_3, t)) = \min(9, 10, 7, 8) = 7$.

(c) **Augment Flow:** We add 7 units of flow along p_3.

(d) **Total Flow:** $|f| = 16 + 7 = 23$.

Residual Graph G_{f_3}: The augmenting path p_3 is shown in red.



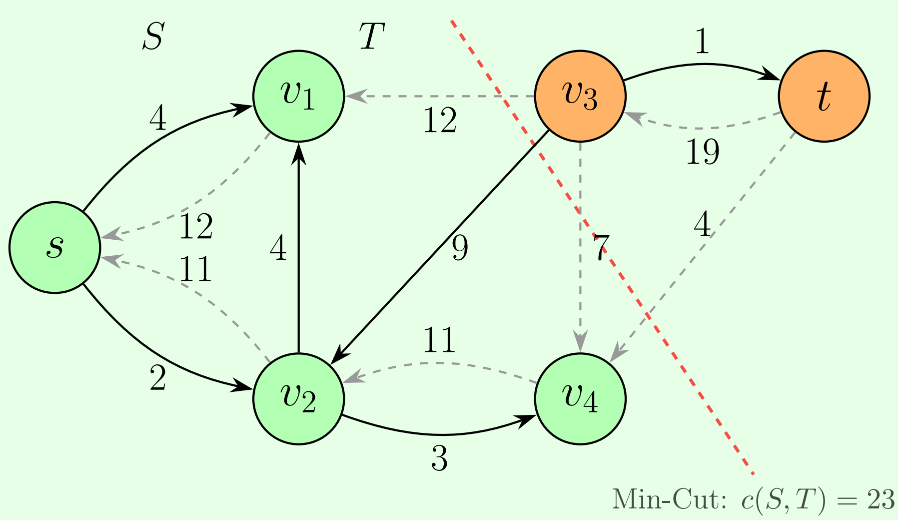
Iteration 4 (Termination)

(a) **Find Path (BFS):** Run BFS on G_{f_3}.
- Saturated forward edges: (v_1, v_3), (v_4, t), (v_4, v_3).
- We can search: $s \to v_1$, $s \to v_2$. From v_2, we can go $v_2 \to v_4$.
- From v_4, all forward edges are saturated. $c_f(v_4, v_3) = 0$, $c_f(v_4, t) = 0$.
- The set of reachable nodes is $S = \{s, v_1, v_2, v_4\}$.

The sink t is not reachable. No $s - t$ path exists in the residual graph.

(b) **Algorithm Terminates.**

Final Residual Graph G_{f_3}: The s-side of the min-cut, S, is shown in green.



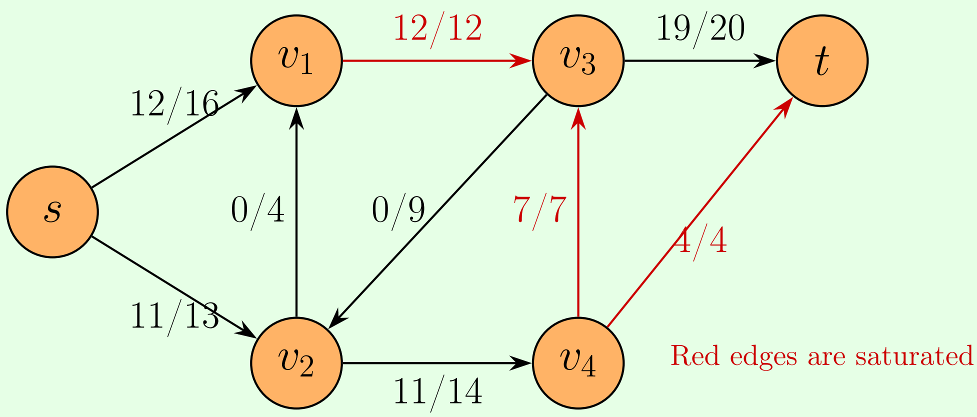
Final Result

The algorithm terminates. The maximum flow is the total flow from the last iteration.

$$\textbf{Maximum Flow} = \mathbf{23}$$

The final flow f is the sum of the augmentations. The min-cut is (S, T) where $S = \{s, v_1, v_2, v_4\}$ and $T = \{v_3, t\}$. The capacity of the cut is $c(v_1, v_3) + c(v_4, v_3) + c(v_4, t) = 12 + 7 + 4 = 23$, which matches the max-flow.

Final Flow Graph (flow/capacity):



3. Longest Augmenting Paths (attributed to CLRS Exercise 24.3-3)

   Let $G = (V, E)$ be a bipartite graph with vertex partition $V = L \cup R$, and let G' be its corresponding flow network. Give a good upper bound on a length of any augmenting path found in G' during the execution of FORD-FULKERSON.

Solution:

By definition, an augmenting path is a simple path $s \rightsquigarrow t$ in the residual network G'_f. Since G has no edges between vertices in L and no edges between vertices in R, neither does the flow network G' and hence neither does G'_f. Also, the only edges involving s or t connect s to L and R to t. Note that although edges in G' can go only from L to R, edges in G'_f can also go from R to L. Thus any augmenting path must go

$$s \to L \to R \to \cdots \to L \to R \to t$$

crossing back and forth between L and R at most as many times as it can do so without using a vertex twice. It contains s, t, and equal numbers of distinct vertices from L and R—at most $2 + 2 \cdot \min(|L|, |R|)$ vertices in all. The length of an augmenting path (i.e., its number of edges) is thus bounded above by $2 \cdot \min(|L|, |R|) + 1$.