**P1**

---

**Prompt:**

Design a polynomial time algorithm to find the minimum vertex cover in a bipartite graph G = (X, Y, E).

---

**Algorithm**

Let FFA(H) be the Ford-Fulkerson algorithm which is given a network H = (X, Y, E) with flow capacity c, a source node s, and a sink node t and returns the minimum s-t cut (A, B).

```
Input: The bipartite graph G = (X, Y, E)
Output: The flow network H
FlowNetwork(G):
    Let H be a copy of G
    For each edge e in H.E
        Let x be the vertex connected to e in H.X
        Let y be the vertex connected to e in H.Y
        Let e be the directed edge (x, y)
        Let e.capacity = ∞
    EndFor
    Add vertex s to H
    For each vertex x in H.X do
        Let e be the directed edge (s, x)
        Let e.capacity = 1
    EndFor
    Add vertex t to H
    For each vertex y in H.Y do
        Let e be the directed edge (y, t)
        Let e.capacity = 1
    EndFor
    Return H

Input: The flow network H = (X, Y, E) and the min s-t cut (A, B) in H
Output: A vertex cover S of G such that |S| = cap(A, B)
VertexCover(H, A, B):
    Let X_B = H.X ∩ B
    Let Y_A = H.Y ∩ A
    Let S = X_B ∪ Y_A
    Return S

Input: The bipartite graph G = (X, Y, E) and the min vertex cover S of G
Output: The min s-t cut (A, B) in H
MinCut(G, S):
```

```
    Let H = FlowNetwork(G)
    Let A = H.s ∪ (H.X \ S) ∪ (H.Y ∩ S)
    Let B = H.t ∪ (H.X ∩ S) ∪ (H.Y \ S)
    Return (A, B)

Let C = FFA(H)
Let S = VertexCover(H, C)
Output S
```

---

**Claim:**

The algorithm terminates in polynomial time.

**Proof:**

That FlowNetwork terminates in polynomial time is easily seen, given that each loop iterates over finite, monotonically decreasing sets. Therefore, FlowNetwork terminates in O(|X| + |Y| + |E|) which is polynomial. That VertexCover and MinCut terminate in constant time is obvious. We know that Ford-Fulkerson terminates in polynomial time from HW7 P3. Thus, the the algorithm terminates in polynomial time.

---

**Claim:**

Given a bipartite graph G = (X, Y, E), the algorithm outputs the minimum vertex cover of G.

**Proof:**

```
From lecture 23 FlowNetwork algorithm is correct
    Let flow network H = (X, Y, E) be the output of FlowNetwork(G)

From lecture 23 Ford-Fulkerson algorithm is correct
    Let min s-t cut (A, B) be the output of FFA(H)
    cap(A, B) = cap(min cut)

Let S = (H.X ∩ B) ∪ (H.Y ∩ A).

S is a vertex cover
    Let X_A = X ∩ A
    Let X_B = X ∩ B
    Let Y_A = Y ∩ A
    Let Y_B = Y ∩ B
    There are no edges between X_A and Y_B
```

```
    For every vertex xa in X_A
        For every edge e between X_A and Y_A
            Let ya be the vertex connected to e in Y_A
            ya is in S
    For every vertex ya in Y_A
        For every edge e incident to ya
            ya is in S
    For every vertex xb in X_B
        For every edge e incident to xb
            xb is in S
    For every vertex yb in Y_B
        For every edge e incident Y_B and X_B
            let xb be the vertex connected to e in X_B
            xb is in S
    Every edge between X and Y has a vertex in S
    thus S is a vertex cover

|S| = cap(A, B)
    |S| = |X_B| + |Y_A|
    cap(A, B) = |X_B| + |Y_A|
    thus |S| = cap(A, B)

Since |S| = cap(A, B)
    |S| = cap(min cut)

There are two possibilities
    Either |S| = |min vertex cover|
    Or |S| != |min vertex cover|

If |S| = |min vertex cover| then
    since |S| = cap(min cut)
    |min vertex cover| = cap(min cut)
Else then
    |min vertex cover| !> |S|
    So |min vertex cover| < |S|
    Then |min vertex cover| < cap(min cut)
        since |S| = cap(min cut)
        |min vertex cover| < cap(min cut)

Thus, given min s-t cut (A, B)
    We find vertex cover S such that
        |S| = cap(A, B) which implies that
        |S| = cap(min cut)
    and
        |min vertex cover| ≤ |S| which implies that
        |min vertex cover| ≤ cap(min cut)

Reset previously used variables S, A, B

Now, suppose |S| = |min vertex cover|
```

```
Let (A, B) be an s−t cut such that
    Let A = H.s ∪ (H.X \ S) ∪ (H.Y ∩ S)
    Let B = H.t ∪ (H.X ∩ S) ∪ (H.Y \ S)

(A, B) is an s−t cut since
    s is in A
    and t is in B
    thus (A, B) is an s−t cut

cap(A, B) = |S|
    cap(A, B) = |X_B| + |Y_A|
    |S| = |X_B| + |Y_A|
    thus cap(A, B) = |S|

Since cap(A, B) = |S|
    cap(A, B) = |min vertex cover|

There are two possibilities
    either cap(A, B) = cap(min cut)
    or cap(A, B) != cap(min cut)

If cap(A, B) = cap(min cut) then
    since cap(A, B) = |min vertex cover|
    cap(min cut) = |min vertex cover|
Else then
    cap(min cut) !> cap(A, B)
    So cap(min cut) < cap(A, B)
        Since cap(A, B) = |min vertex cover|
        then cap(min cut) < |min vertex cover|

Thus, given min vertex cover S
    we find s−t cut (A, B) such that
        cap(A, B) = |S| which implies that
        cap(A, B) = |min vertex cover|
    and
        cap(min cut) ≤ cap(A, B) which implies that
        cap(min cut) ≤ |min vertex cover|

Combining the above, show algorithm finds min vertex cover from G
    Reset all variables
    Let bipartite graph G = (X, Y, E)
    Let flow network H be output of FlowNetwork(G)
    Let min s−t cut (A, B) be output of FFA(H)
    min cut (A, B) is necessary and sufficient for min vertex cover S
        since cap(min cut) ≤ |min vertex cover| ≤ cap(min cut)
    which implies that
        cap(min cut) = |min vertex cover|
    Therefore create min vertex cover S from min cut (A, B)
    Thus, the algorithm finds min vertex cover S from G
```