P3

Prompt:

Given the described graph G, design an algorithm that runs in time polynomial in n and $\max_e c(e)$ and outputs "yes" if we can satisfy all demands and "no" otherwise.

Algorithm:

Let Abs(x) be the standard absolute value function that is given a real number x and returns x if $x \geq 0$ and -x if $x < 0$.

Let FFA(G) be the Ford-Fulkerson algorithm which is given a network G = (V, E) with flow capacity c, a source node s, and a sink node t and returns the flow f from s to t of maximum value.

```
Input: The described graph G
Output: "yes" if all demands can be satisfied and "no" otherwise
DemandSatisfied(G):
    Let H be a copy of G
    Add vertex s to H
    Add vertex t to H
    Set r(s) = 0
    Set r(t) = 0
    Let demand = 0

    For each vertex v in H do
        If r(v) > 0 then
            Add edge e from s to v with c(e) = r(v)
        Else if r(v) < 0 then
            Add edge e from v to t with c(e) = Abs(r(v))
            demand += Abs(r(v))
        EndIf
    EndFor

    Let maxFlow = FFA(H)
    If maxFlow >= demand then
        return "yes"
    Else then
        return "no"
    EndIf
```

Claim:

The algorithm terminates in time polynomial in n and $\max_e c(e)$.

Proof:

Since we assume that the capacities in G are integers, we know that Ford-Fulkerson is bounded by O(n * m * $\max_e c(e)$) and hence has the same bound on H. The algorithm loop visits each vertex exactly once and so is bounded by the number of vertices in H. Therefore, the Ford-Fulkerson runtime dominates the loop runtime. Thus, the algorithm terminates in O(n * m * $\max_e c(e)$).

Claim:

The algorithm outputs "yes" if and only if we can supply all the demands and "no" otherwise.

Proof:

First, consider the construction of the graph H within the algorithm. Let H be a graph that is a copy of the given arbitrary graph G with the following additions. Let H have two additional vertices s and t where r(s) = 0 and r(t) = 0. For each vertex v in H such that r(v) > 0, let s have a directed edge e to v and let the capacity c(e) = r(v). For each vertex u in H such that r(u) < 0, let u have a directed edge f to t and let the capacity c(f) = Abs(r(u)).

Note the following facts about the construction of H:

1. The Sum{c(e) for all edges e directed from s} = Sum{r(v) for all vertices v with r(v) > 0 in H} = the sum of all sources in G = the supply in G.
2. The Sum{c(f) for all edges f directed to t} = Sum{Abs(r(u)) for all vertices u with r(u) < 0 in H} = the positive sum of all sinks in G = the demand in G.

Now, we prove the claim by considering the biconditional from both directions.

If the algorithm is given a graph G with feasible supply/demand then the algorithm outputs "yes". For a contradiction, assume that G has feasible supply/demand and the algorithm outputs "no"". By the construction of H, we know that since G has a feasible supply/demand that:

1. The sum of the capacity of edges leaving s is ≥ the sum of the capacity of the edges entering t and
2. there is sufficient capacity between s and t to carry the current from s to t.

Therefore, the maximum flow found by the Ford-Fulkerson algorithm on H will be ≥ the demand of G and the algorithm outputs "yes". Which is a contradiction.

And from the other direction.

If the algorithm outputs "yes" then it was given a graph G with feasible supply/demand. For a contrapositive, assume that G does not have feasible supply/demand that the algorithm outputs "no". By construction of H, we know that since G does not have feasible supply/demand that:

1. The sum of the capacity of edges leaving s is less than the sum of the capacity of the edges entering t or
2. there is insufficient capacity between s and t to carry current from s to t.

Therefore, the maximum flow found by the Ford-Fulkerson algorithm on H will be $<$ the demand of G and the algorithm outputs "no".

Since we've shown both directions of the bidirectional, the claim holds. $\square$