PSET 5 wdc22

3

```
def transform(G=(V,E,c), s, t, F)
    let d: V → ℝ be a dictionary of vertex demands
    d[v] = 0 ∀ v ∈ V
    for e=(u,v) ∈ E s.t. c[e] > 0:
        let t be a vertex
        V* = {V ∪ {t}}
        d[t] = -c[e]
        d[v] = d[v] + c[e]
        E* = {E - {(u,v)}}
        E* = {E ∪ {(u,t), (v,t)}}
    d[s] = -F
    d[t] = F
    return G=(V,E), d
```

The algorithm runs in $O(n)$, as it only iterates over the input once. It is correct as it simulates edge capacity by introducing an interstitial vertex for each edge $e \in E$ s.t. $c[e] > 0$, and assigning that vertex a supply equal to e's capacity, then giving $(e=(u,v))$ v an equal demand. So, when flow is pushed onto the edge, the demand-flow treats it as max-flow would treat capacity. Also, s is set to supply F flow and t to demand F flow, thus satisfying the F-target flow constraint