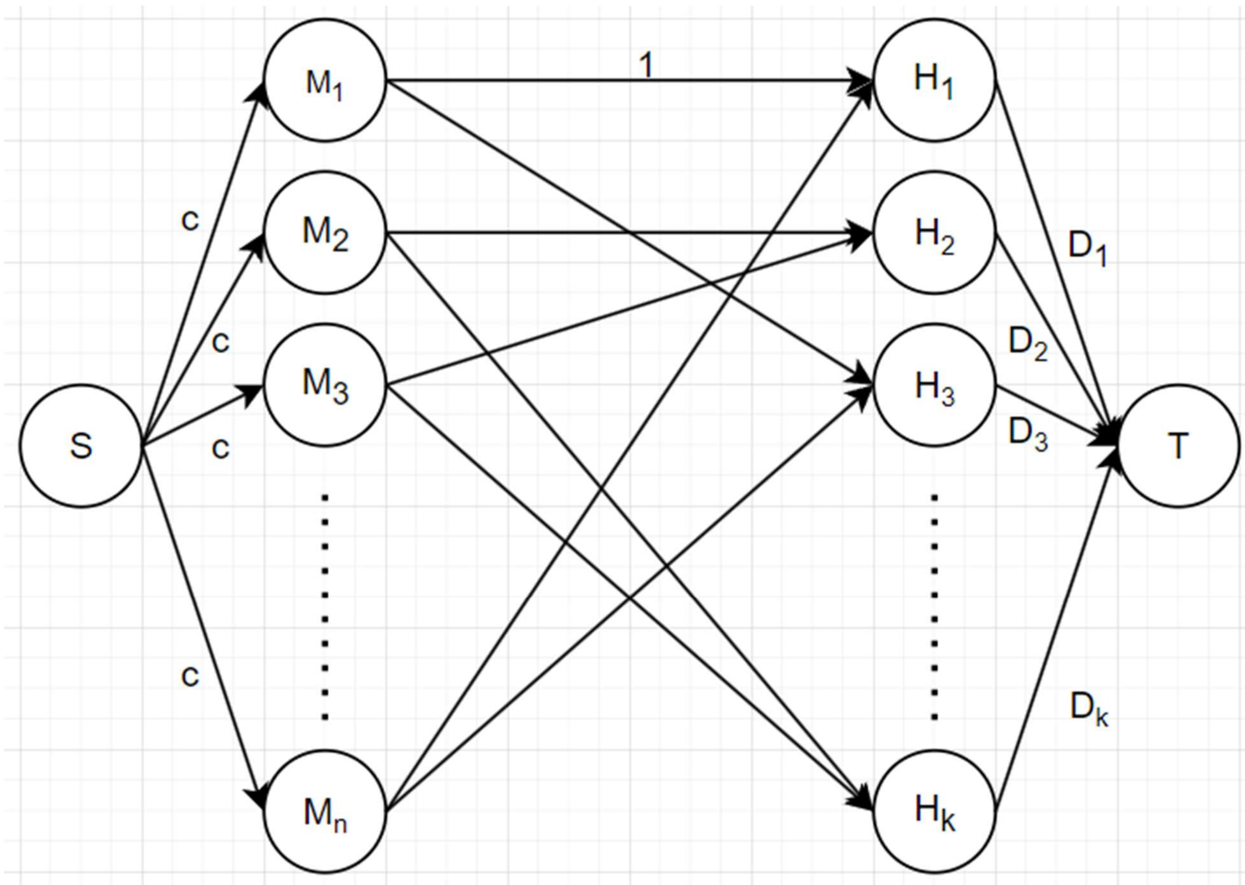
**Problem 4**

Consider the following graph (all numbers on edges represent the capacity of the edge, not the flow):




$M_i$ represents doctor i, and $H_j$ represents vacation/holiday j. The flow of this graph is the vacation/holiday days that can be worked. Each doctor can only work at most c vacation days, so each edge connecting S to a doctor node $M_i$ has capacity c. Each doctor can only work 1 day per vacation period, so each edge from a doctor node $M_i$ to a vacation node $H_j$ has capacity 1. Note that the amount of connections between a doctor node $M_i$ and the set of vacation nodes H, is related to the doctor's availability. A doctor node will only flow into vacation nodes if the doctor has availability on that vacation. Each edge connecting vacation node $H_j$ to T has capacity $D_j$ where $D_j$ is the number of days in that vacation period. The flow from $H_j$ to T represents the number of days in that vacation period that can be covered by a doctor. If the flow into T is equal to N, it means that all vacation days were able to be covered.

**Pseudocode Algorithm (Ford-Fulkerson):**

#Setting up graph, this code will create the graph pictured on the previous page

```
create graph G=(V,E)
add source node S to G
add sink node T to G
add nodes Mi to G for all i ∈ n
add nodes Hj to G for all j ∈ k
add edge (S, Mi) with capacity c to G for all i ∈ n
add edge (Hj, T) with capacity Dj to G for all j ∈ k
for every node Mi do
        for every vacation period Hj that doctor Mi is available for do
                add edge (Mi, Hj) with capacity 1 to G
#Solving part of algorithm, this code will maximize the flow given the pictured graph:
for each edge (u,v) ∈ E do
        (u, v).f = 0
while there exists an s, t path p in Gf do
        cf(p) = min{ cf(u,v) : (u,v) ∈ p do
        for each edge (u,v) ∈ p do
                if (u,v) ∈ E then
                        (u,v).f += cf(p)
                else
                        (v,u).f -= cf(p)
#checking if solution is valid:
maxFlow = sum((Hj, T).f) for all j ∈ k
if maxflow ≥ N
        return "yes"
else
        return "no"
```

**Runtime Analysis:**

It was noted in class that the run time of this algorithm is: $O(VE^2)$

The size of V and E can be calculated as:

There is one vertex for each doctor, one for each vacation period, a sink and a source:

$$O(V) \; = \; O(n+k+2) \; = \; O(n+k)$$

There is an edge connecting each doctor to the source, one connecting each vacation period to the sink, and then up to k edges for each doctor to connect a doctor to all vacation periods they are available for, and since there are n doctors:

$$O(E) \; = \; O(n+nk+k) \; = \; O(nk)$$

Putting this all together, the runtime is:

$$O(VE^2) = O\big((n+k)(n^2k^2)\big) = O(n^3k^2 + n^2k^3)$$