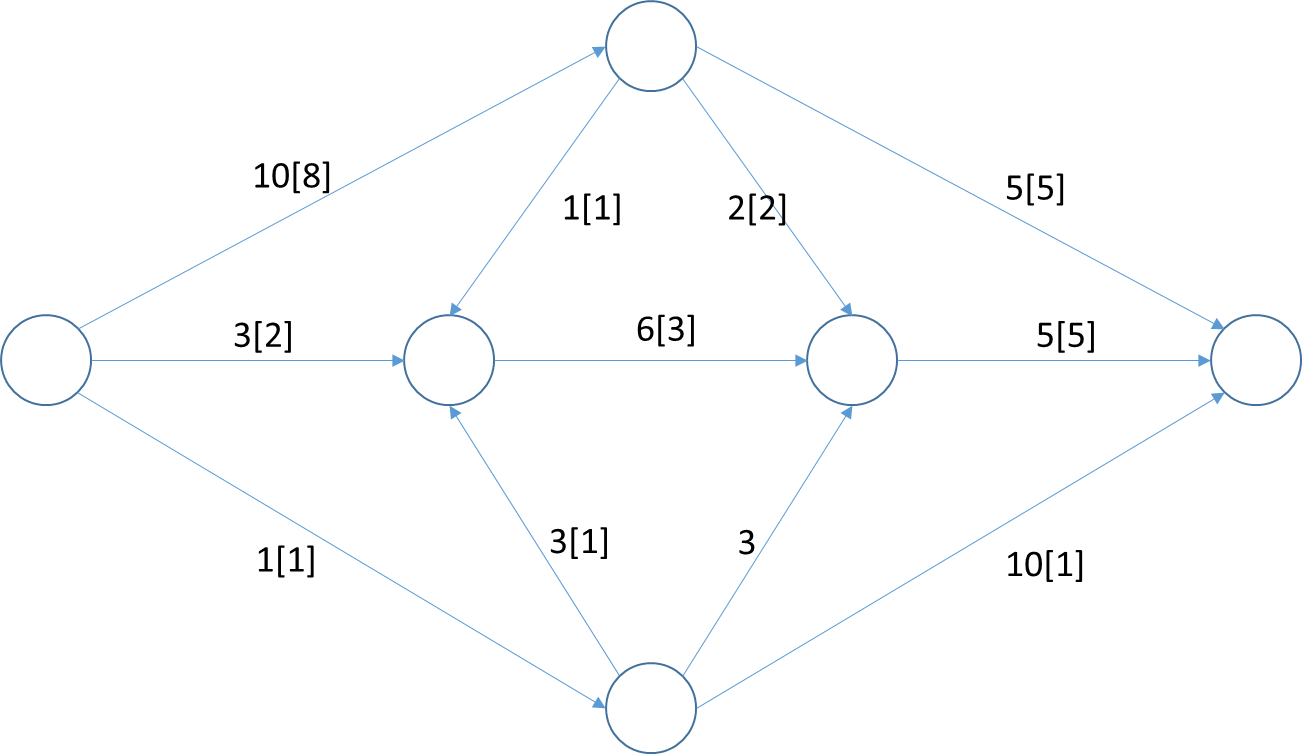
# Exercise 3 p415

## a)

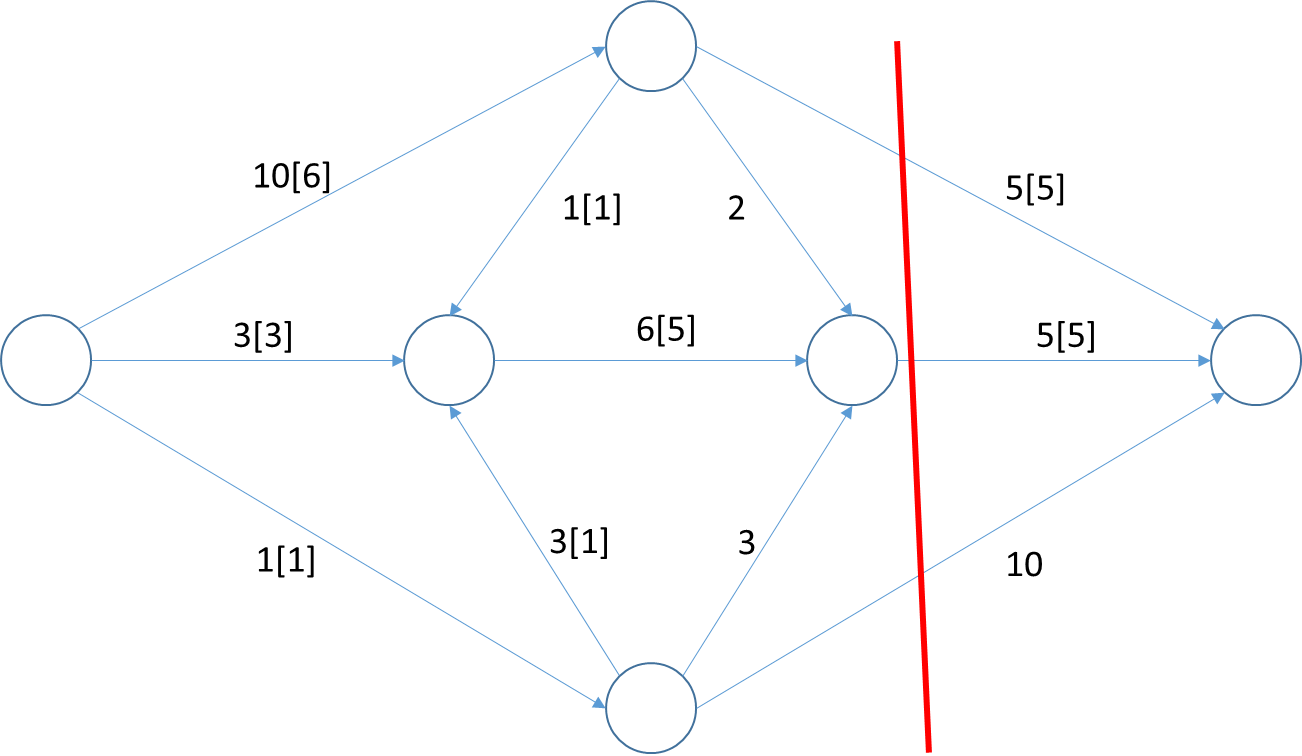
The flow is

It’s not a max flow. Indeed using the max flow algorithm we get this:



## b)

The minimum cut is in red and the capacity is



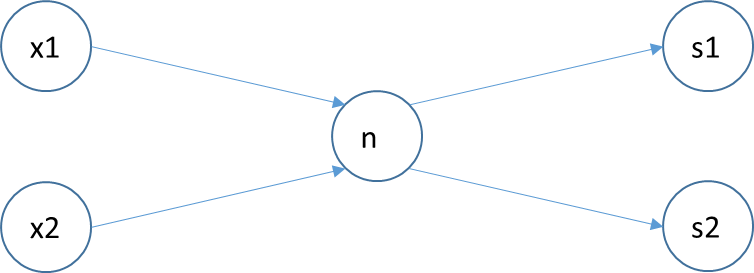
# Exercise 14p421

## a)

We can add a new node s with edges from s to all nodes in X with capacity 1 and add another node t with edges from all nodes in S to t with unlimited capacity. Then we put a capacity of 1 for all other edges. Then we can use Ford-Fulkerson algorithm to solve this problem. Then if the max-flow is equals to the number of nodes in X then we are good. So this is solved in polynomial time

## b)

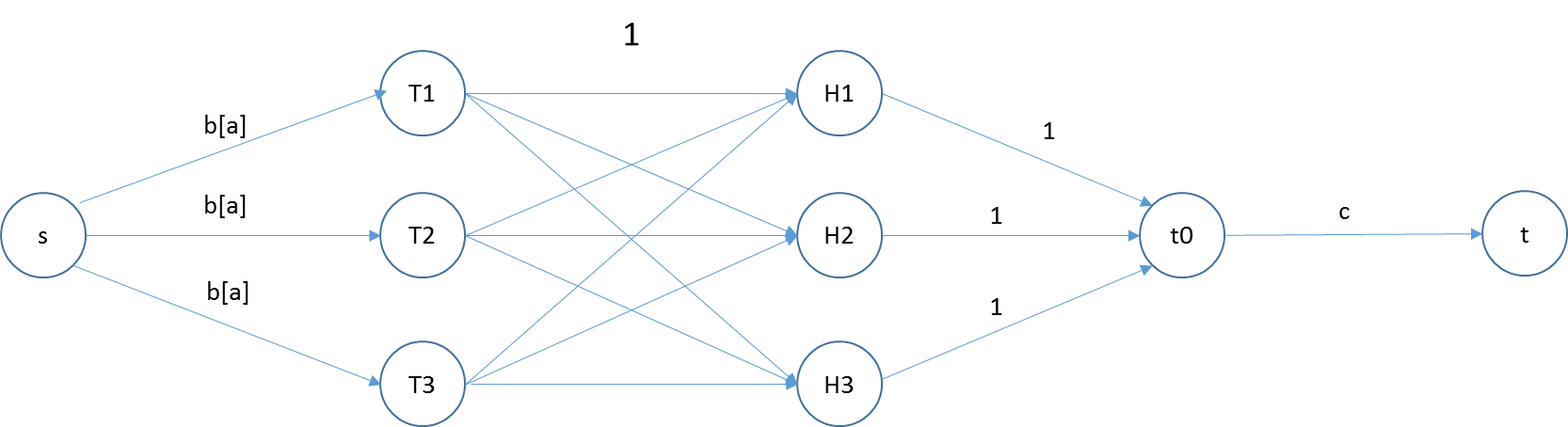
We can take the same approach as before except that we add another step. For all nodes n not in X or S we create 2 nodes n1 and n2 then we take all incoming edges of n and set the destination to n1 and take all outgoing edges and set the origin to n2 and we finally add another edges from n1 to n2 with capacity of 1. We as well put the edges from nodes in S to t with a capacity of 1.



This example show we can have it working with a but not with b as n is in two path

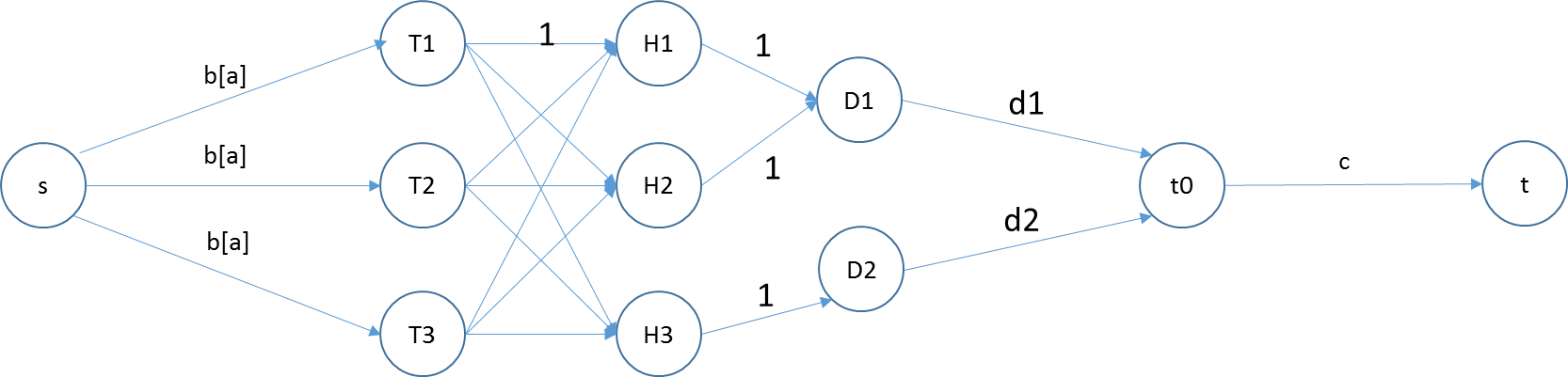
# Exercise 28p432

## a)



We transform this into a flow problem. We have a list of node Tn for the TAs and a list of hours nodes. We have edges from s to each T nodes with max capacity b and minimum a. Then we have edges form each node T to each node H if he can teach at this time with max capacity 1. Then we have edges from each node H to a node with max capacity 1. Finally we have an edge from this node t0 to t with capacity c. Then we can use Ford-Fulkerson to solve this problems. If the algorithm max-flow is different of c then there is no solution to this case.

## b)

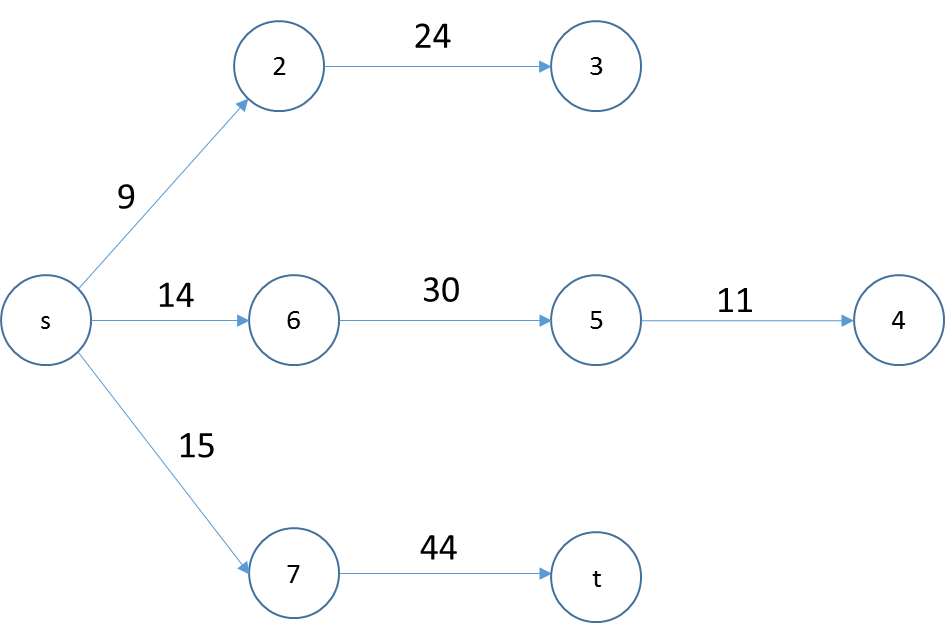


We keep the same way as before except we add another column for days. Each H is going to its corresponding day with a capacity of 1(time only used once) Then from each day we add en edges to t0 with the capacity dn of the day.

## c)

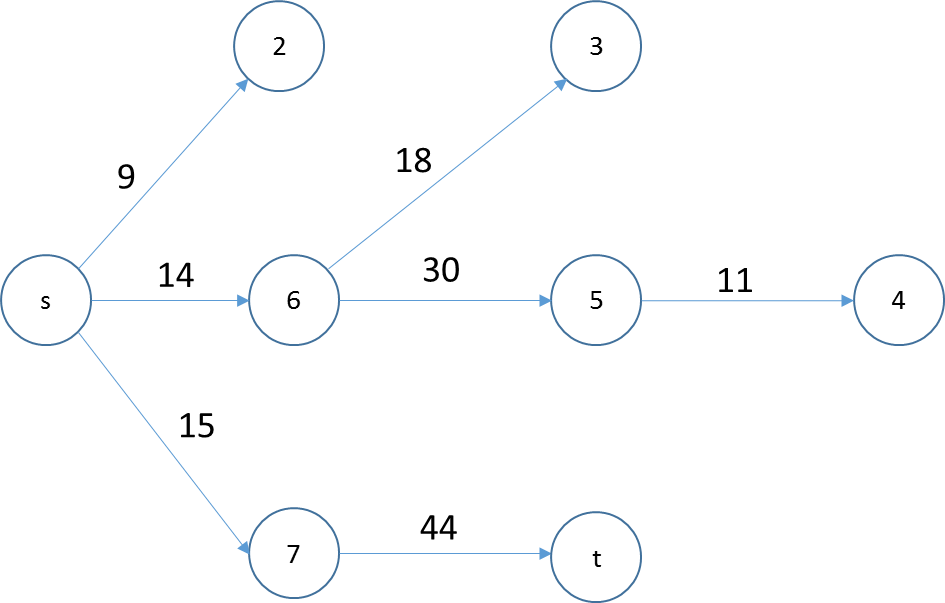
* For a) working we can have T1 and T2 with a=b=1 and c=2 and H1 and H2
* For a) not working we can have T1 and T2 with a=1, b=2 and c=3 and H1, H2. We are missing a hour here
* For b) working we can have T1 and T2 with a=1, b=2, c=4, D1: H1, H2 and D2: H3, H4 and d1=2 and d2 = 2 then we can fit easily here supposing T1 and T2 can teach during those times
* For b) not working we can have T1 and T2 with a=1, b=2, c=4, D1: H1, H2 and D2: H3, H4 and d1=2 and d2 = 3 then it’s not possible to have three hours on day 2

# Exercise



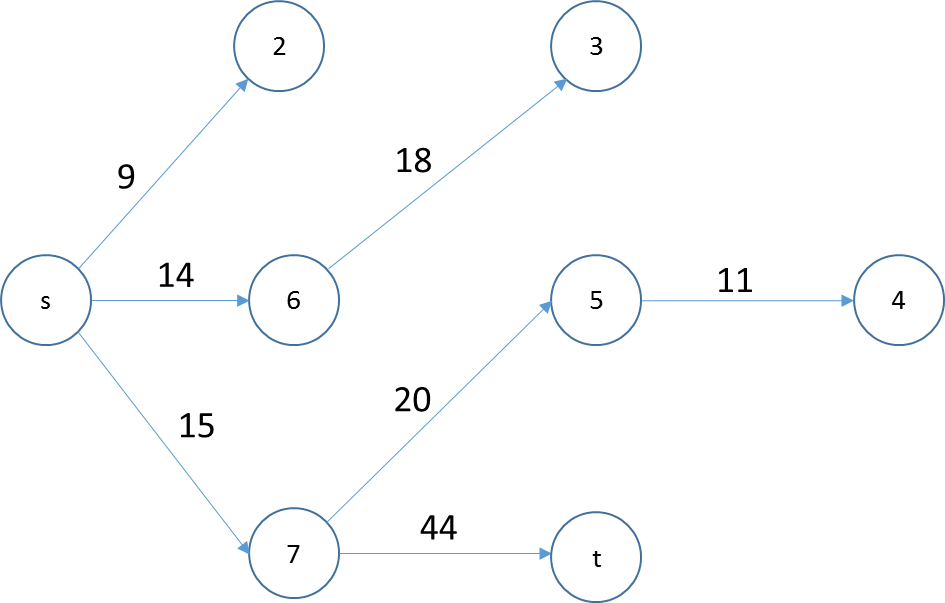
Compute:

* 2: 9
* 3: 33
* 6: 14
* 5: 44
* 4: 55
* 7: 15
* T: 59



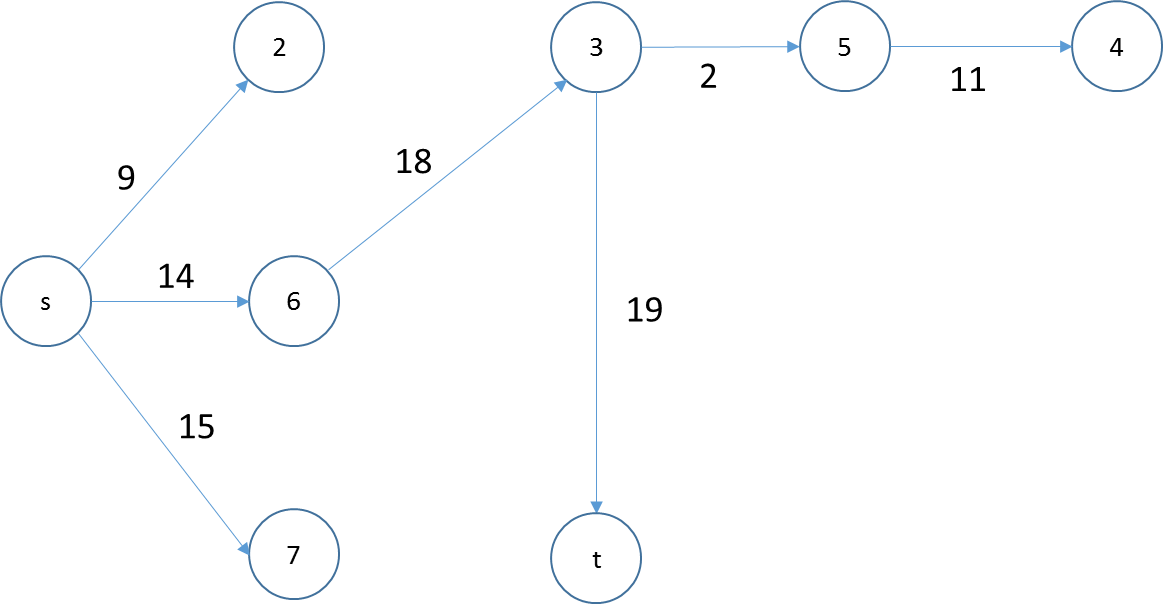
Compute:

* 3: 32



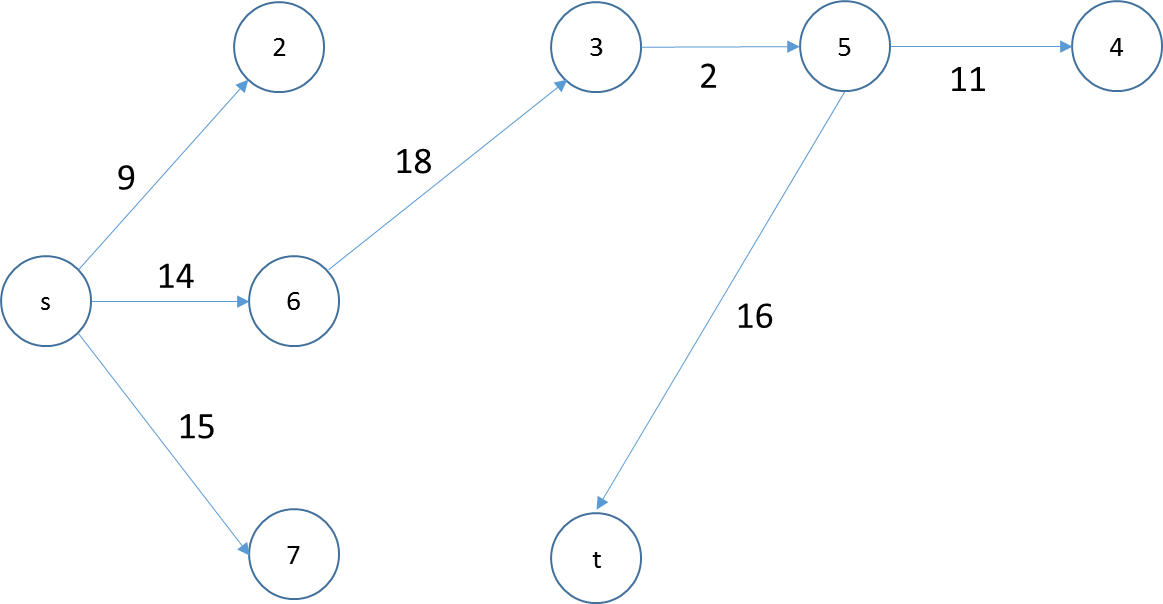
Compute

* 5: 35
* 4: 46



Compute

* 5: 34
* 4: 45
* t: 51



Compute

* t: 50

Done minimum is s635t