Flow Networks
CSCI 432

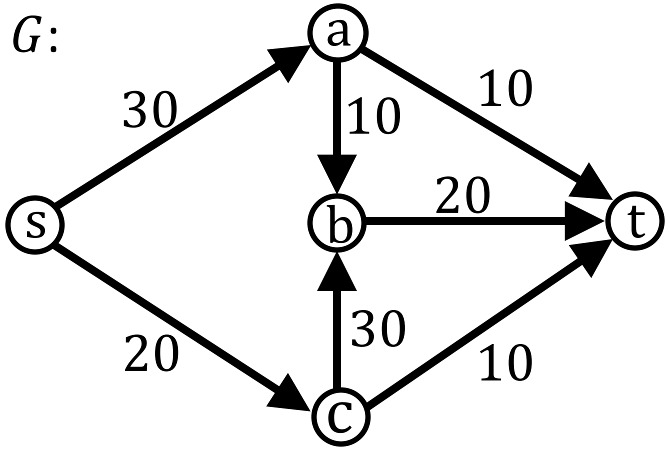
Ford-Fulkerson Algorithm

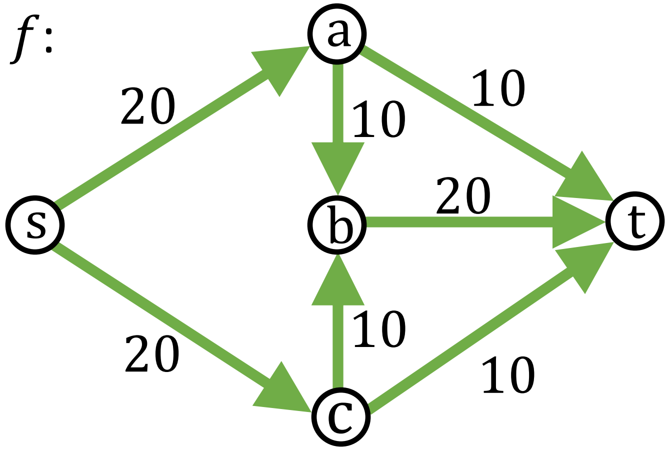
Residual graph for flow f, G_f:

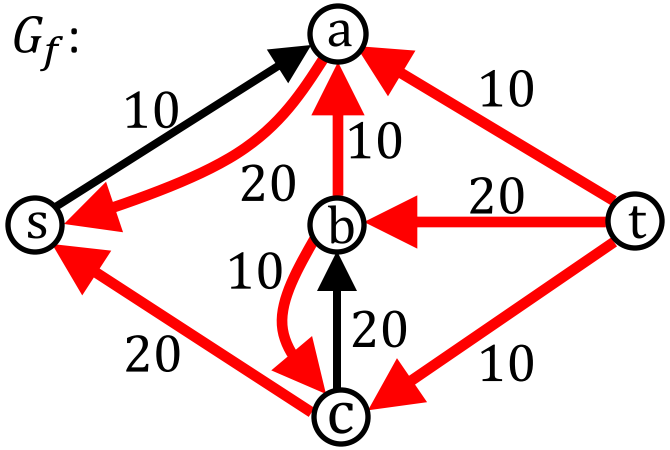
- $\forall e$, if $f(e) < c_e$, let $c_e = c_e - f(e)$.
- $\forall e = (u, v)$, if $f(e) > 0$, create $e' = (v, u)$ with $c_{e'} = f(e)$

```
Max-Flow(G)
   f(e) = 0 for all e in G
   while s-t path in Gf exists
      P = simple s-t path in Gf
      f'= augment(f, P)
      f = f'
      Gf = Gf'
   return f
```

```
augment(f, P)
   b = bottleneck(P,f)
   for each edge (u, v) in P
      if (u, v) is a back edge
         f((v, u)) -= b
      else
         f((u, v)) += b
   return f
```







Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Apollo 13 Filter Problem:

"We need to fit this into the hole for this, using nothing but that"

Work Scheduling

<u>Problem:</u> We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Apollo 13 Filter Problem:

"We need to fit ~~this~~ into the ~~hole for this~~, using nothing but ~~that~~"

work scheduling problem **Max Flow solver** **valid flow network components**

Work Scheduling

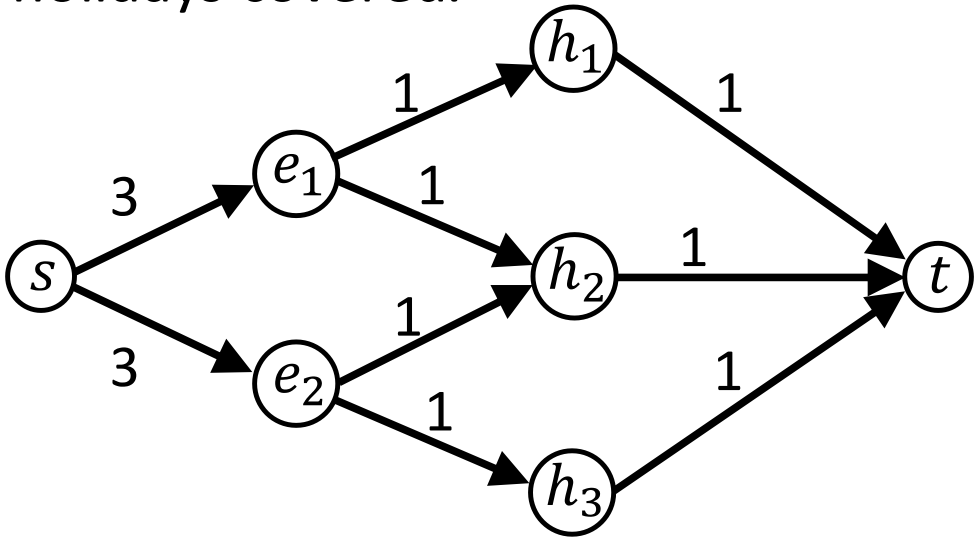
<u>Problem:</u> We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

<u>Algorithm:</u>

??

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.



Algorithm:

1. Build flow network:

a) Make a node for each employee, a node for each holiday, a source, and a sink.

b) Connect the source to each employee node with a capacity of 3.

c) Connect each holiday node to the sink with a capacity of 1.

d) If an employee is able to work a holiday, connect them with a capacity of 1.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

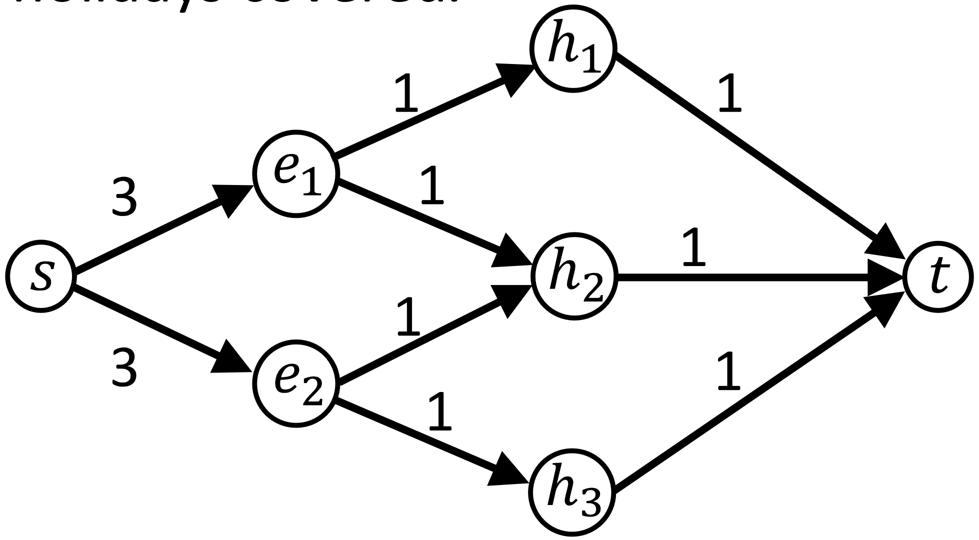
1. Build flow network:

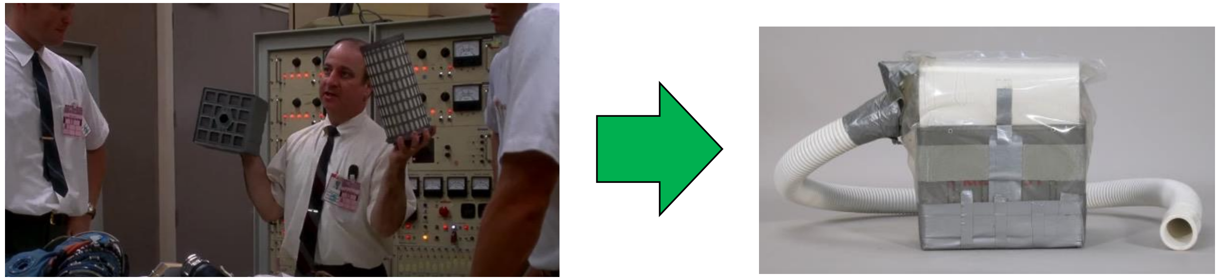
a) Make a node for each employee, a node for each holiday, a source, and a sink.

b) Connect the source to each employee node with a capacity of 3.

c) Connect each holiday node to the sink with a capacity of 1.

d) If an employee is able to work a holiday, connect them with a capacity of 1.



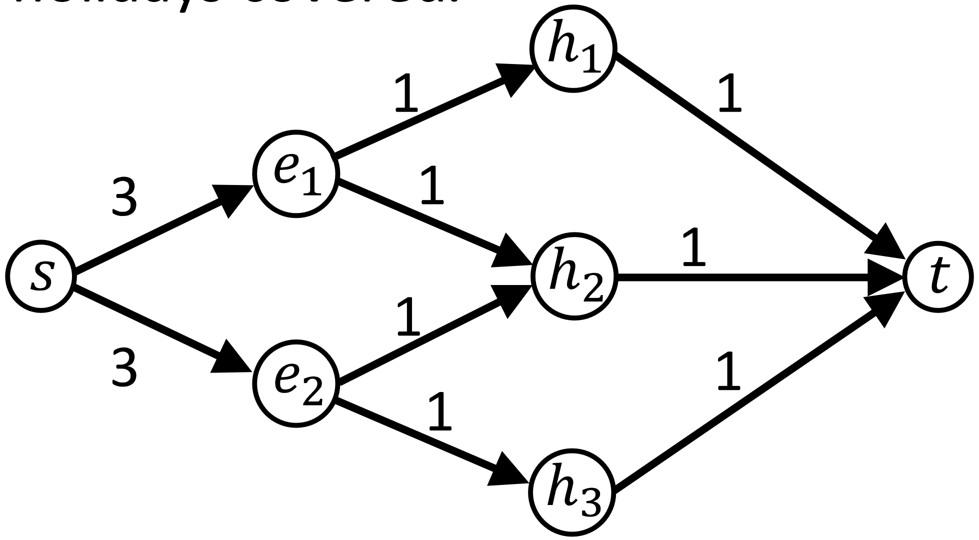


Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. ??

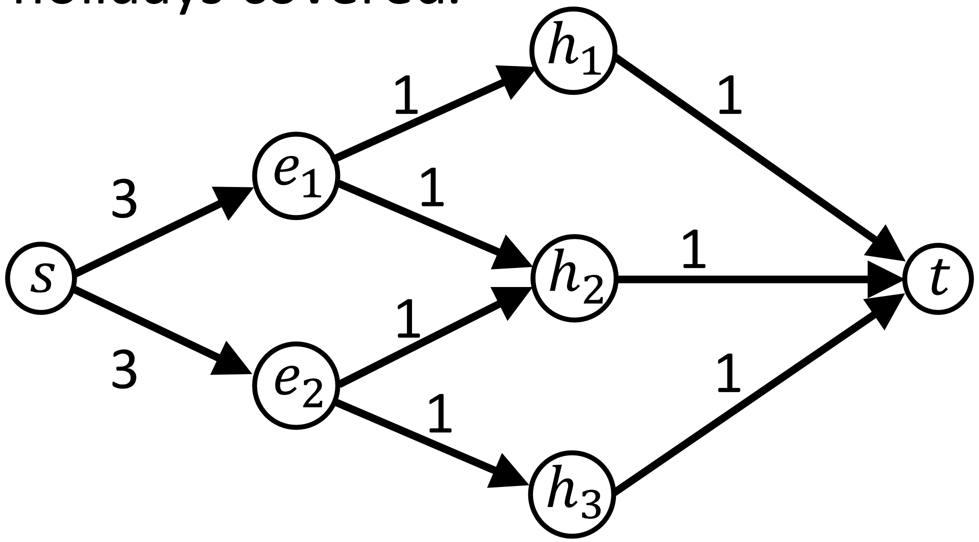


Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.

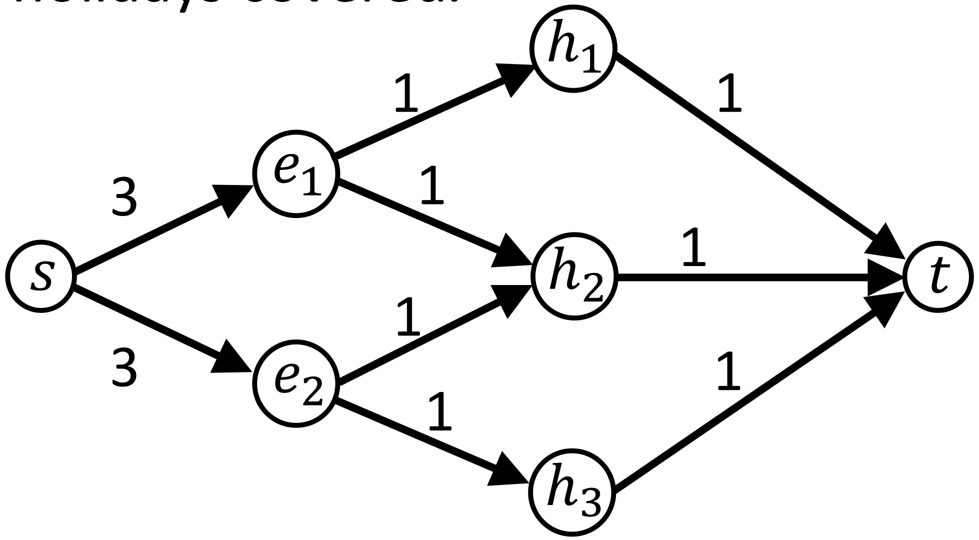


Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. ??

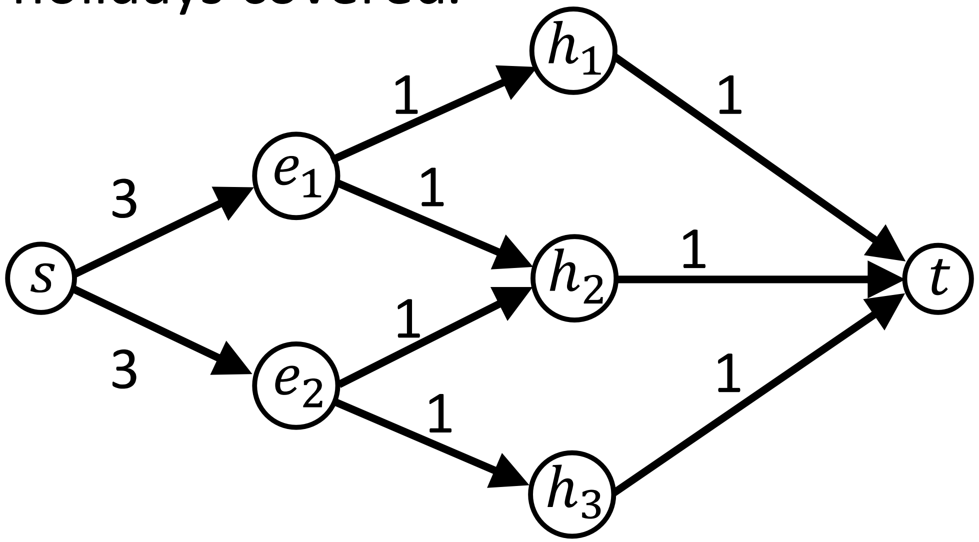


Work Scheduling

<u>Problem:</u> We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

<u>Algorithm:</u>

1. Build flow network.
2. Find max flow.
3. <Translate flow to answer>

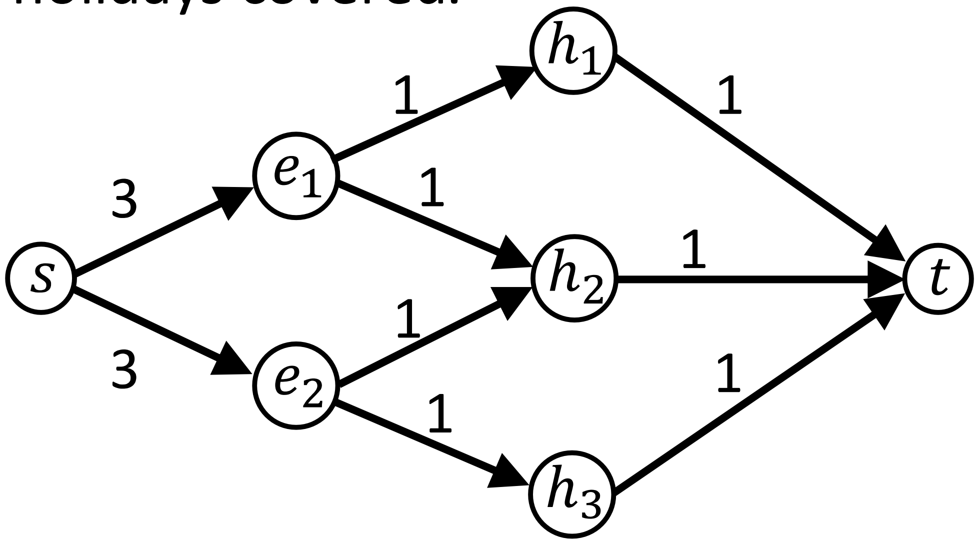


Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

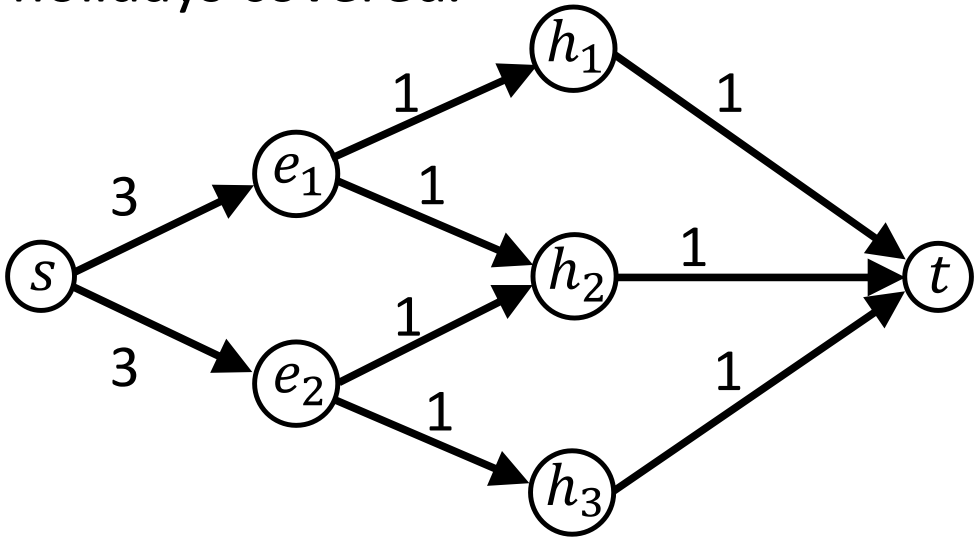


Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



Running Time:

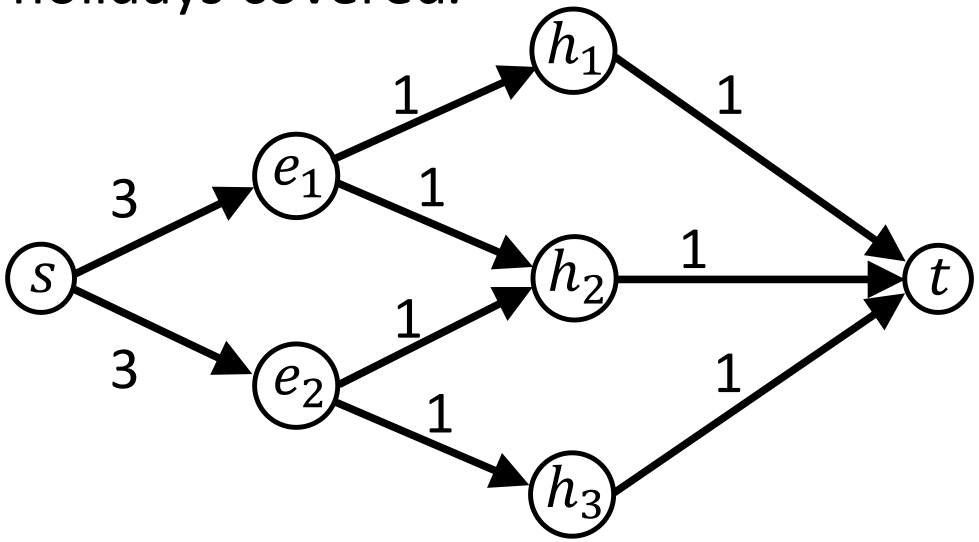
$$O(\text{time to build G}) + O(\text{Max Flow}) + O(\text{time to make answer})$$

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



Running Time:

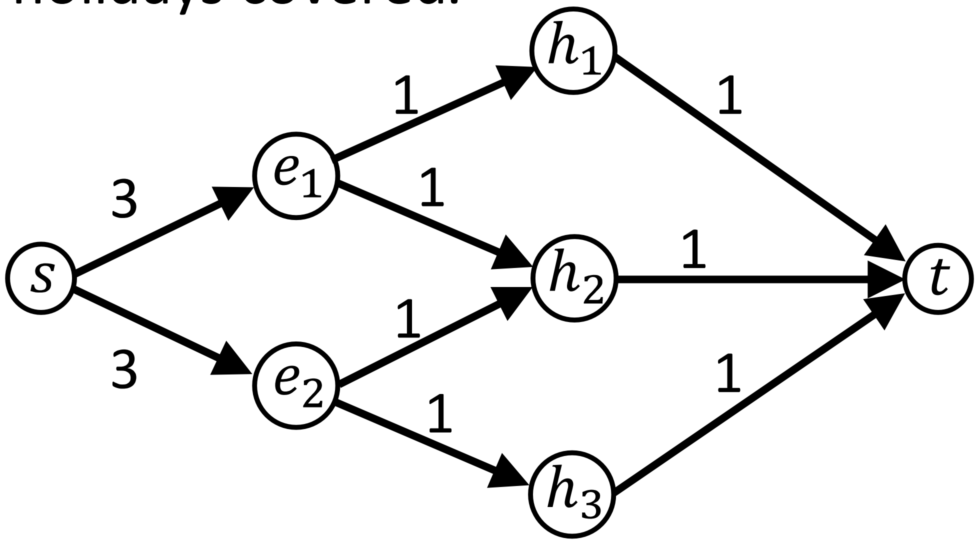
n = # employees, m = # holidays

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



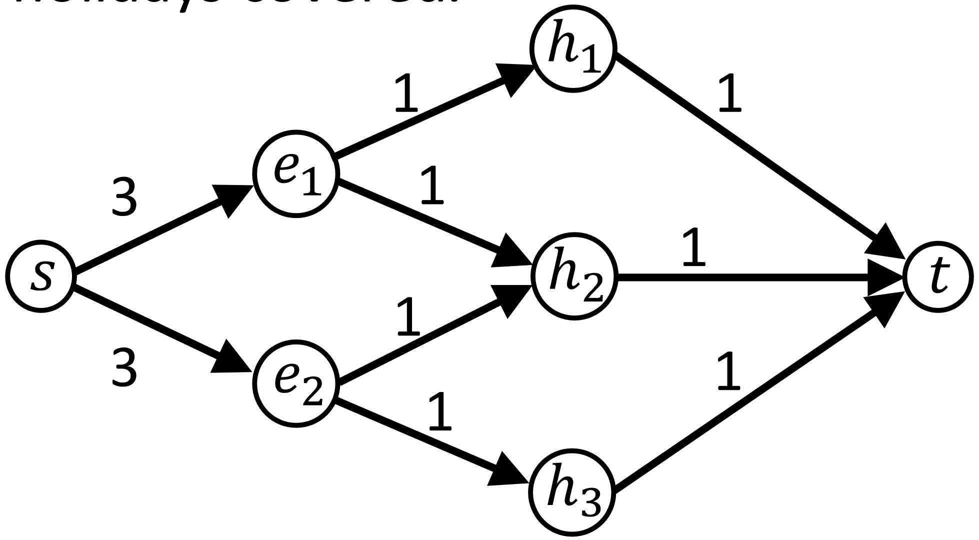
Running Time:

n = # employees, m = # holidays

For $G = (V, E), |G| =$?

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.



Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

Running Time:

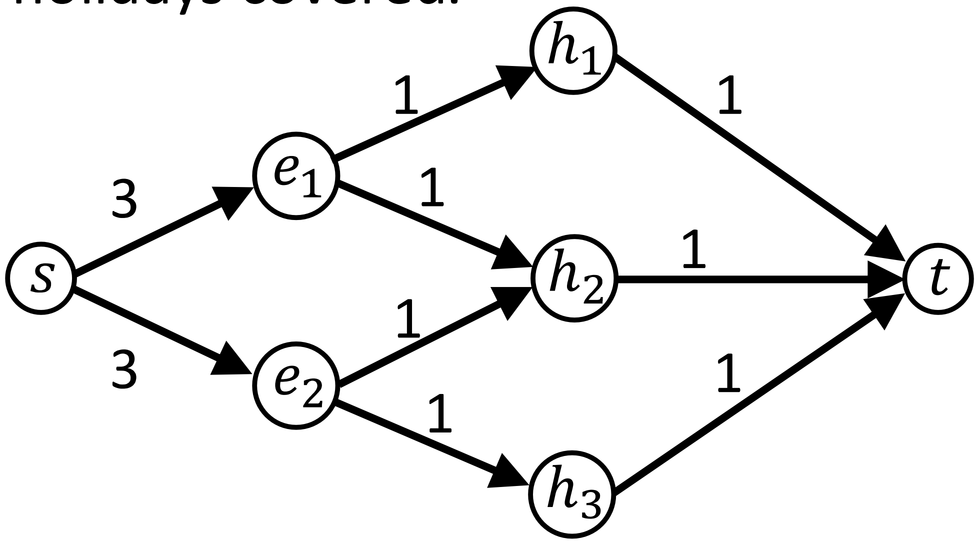
n = # employees, m = # holidays

For $G = (V, E), |V| = n + m + 2, |E| = n + nm + m$

So, building $G \in O(nm)$.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.



Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

Running Time:

n = # employees, m = # holidays

For $G = (V, E), |V| = n + m + 2, |E| = n + nm + m$

So, building $G \in O(nm)$.

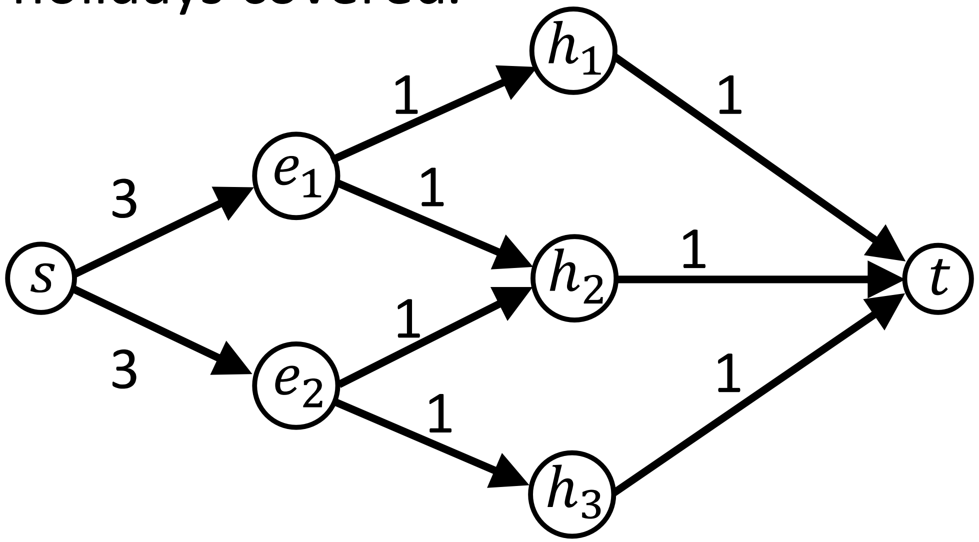
Total running time ∈ ?

Work Scheduling

<u>Problem:</u> We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

<u>Algorithm:</u>

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



<u>Running Time:</u>

n = # employees, m = # holidays

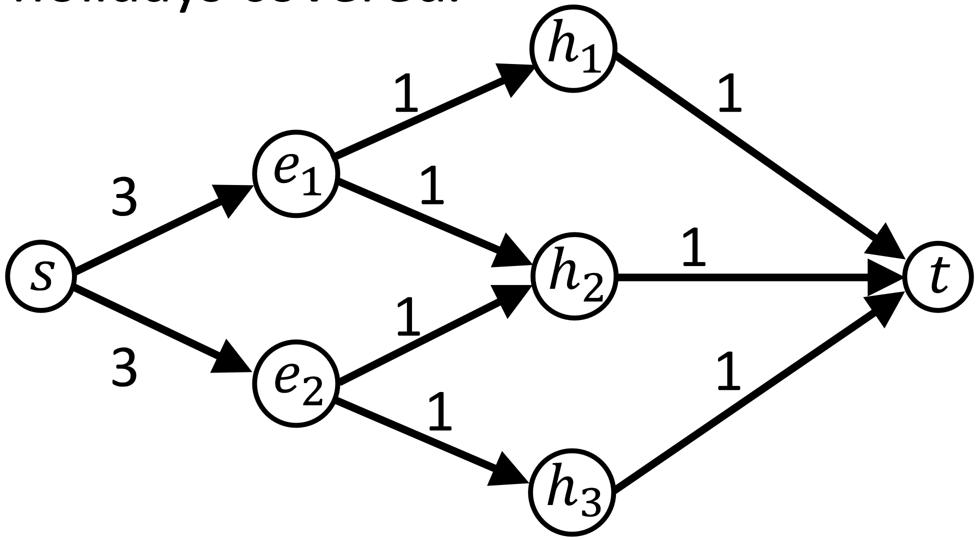
For $G = (V, E), |V| = n + m + 2, |E| = n + nm + m$

So, building $G \in O(nm)$.

Total running time $\in O(nm) + O(\text{Max Flow}) + O(nm) \in O(nm) + O(\text{max flow})$

Work Scheduling

<u>Problem:</u> We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.



<u>Algorithm:</u>

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

<u>Validity:</u>

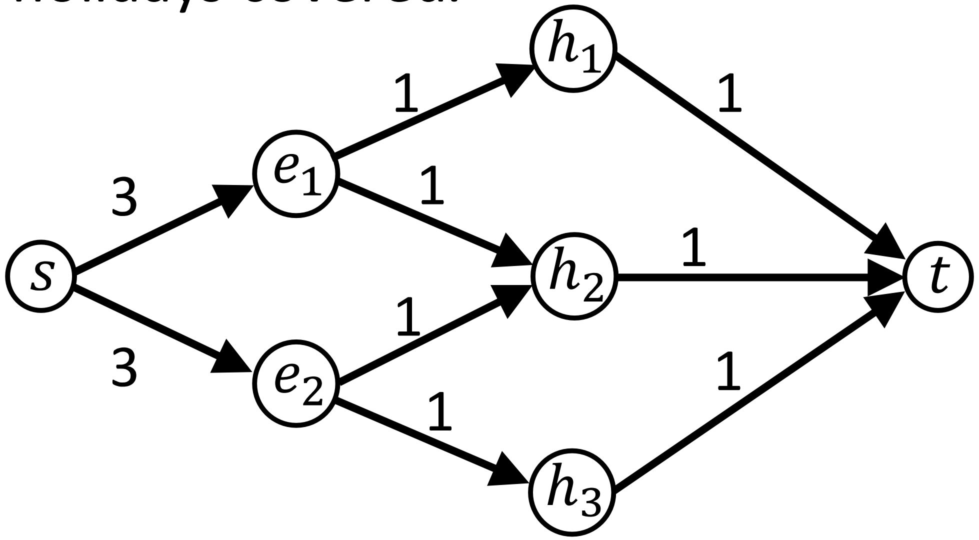
??

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



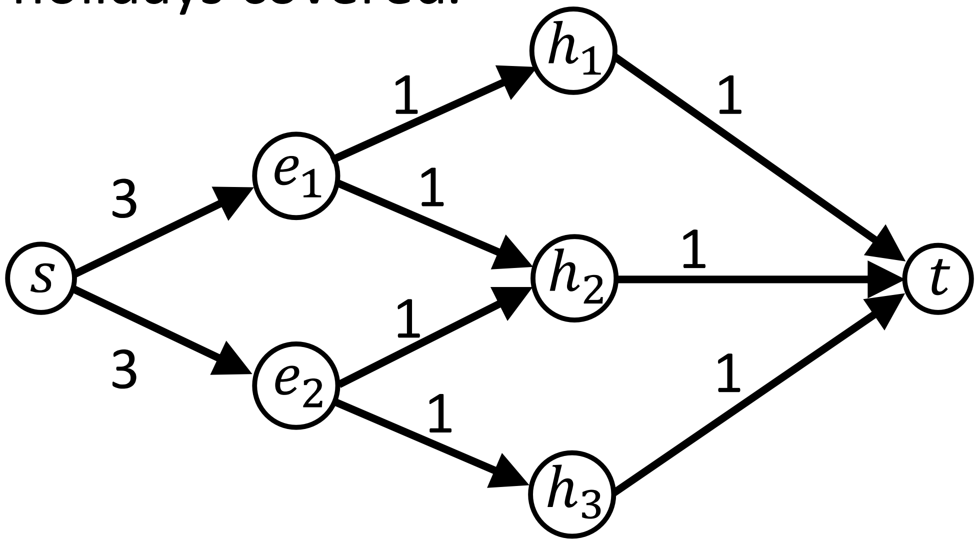
Validity:
Employees cannot be assigned a holiday they are not able to work (no edges exist between incompatible pairs).

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



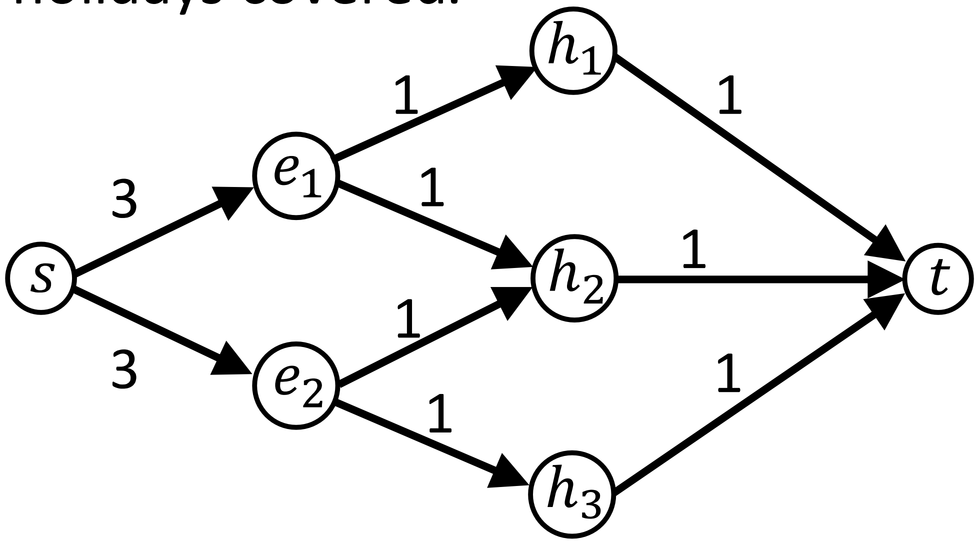
Validity:
Employees cannot be assigned a holiday they are not able to work (no edges exist between incompatible pairs). Employees cannot be assigned to work more than 3 holidays (integer edge capacities yield integer flow values).

Work Scheduling

<u>Problem:</u> We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

<u>Algorithm:</u>

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



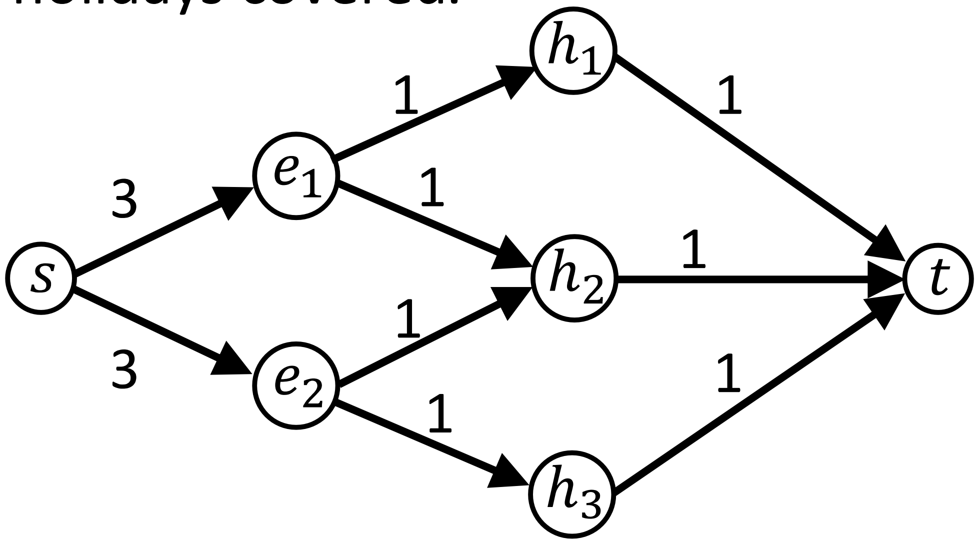
<u>Optimality:</u>

??

We need to show the algorithm's answer is optimal for the problem, not that the max flow found is optimal for the flow network we made.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.



Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

Optimality:

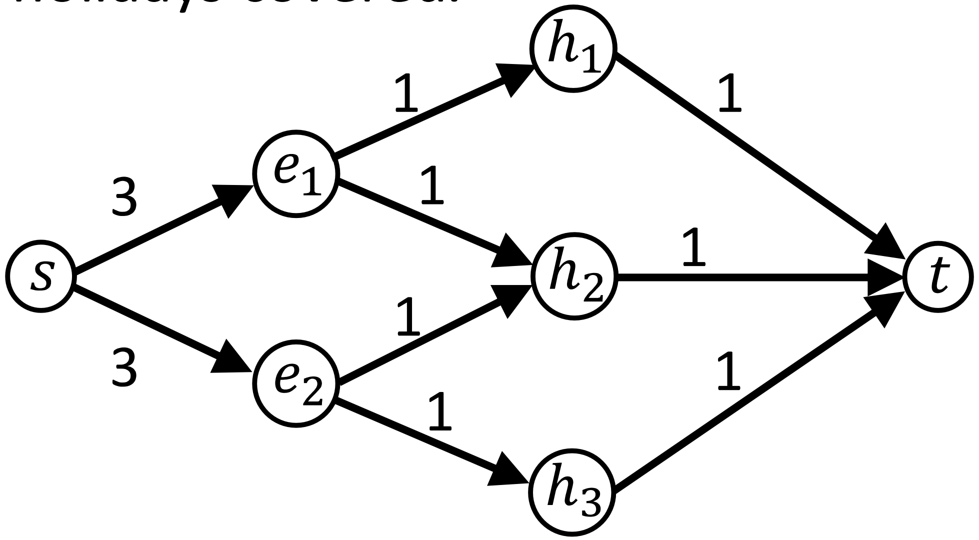
Suppose max flow $F < OPT \leq m$.

\# holidays

Optimal answer to the problem.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.



Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

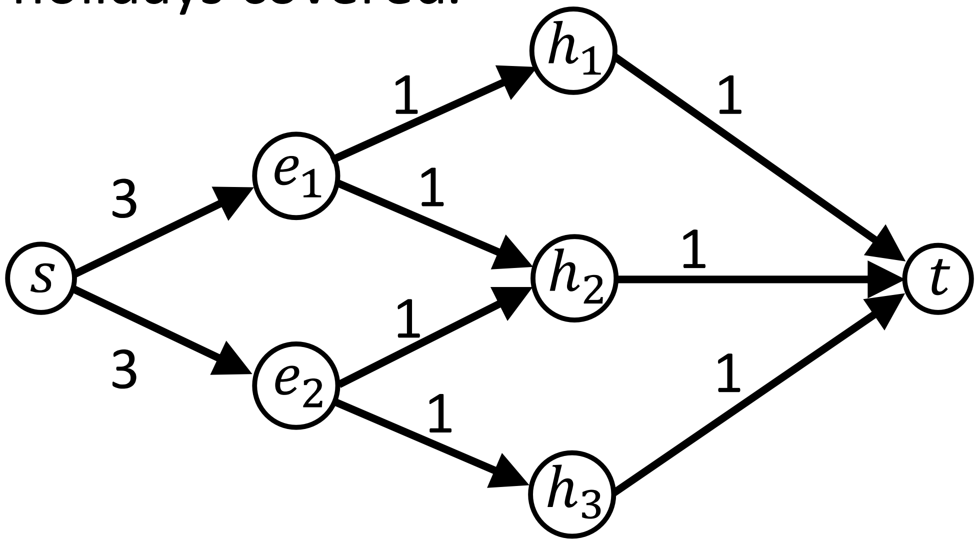
Optimality:

Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays.

Plan: Turn OPT into a valid flow with value larger than F. CONTRADICTION!!!

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.



Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

Optimality:

Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage.

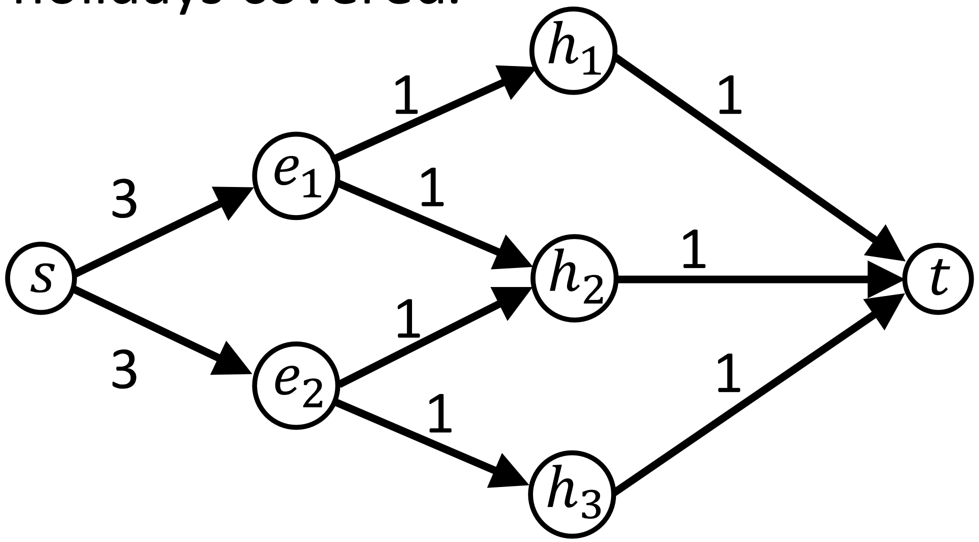
Reducing the number of employees working a single holiday to 1 keeps that holiday covered.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



Optimality:

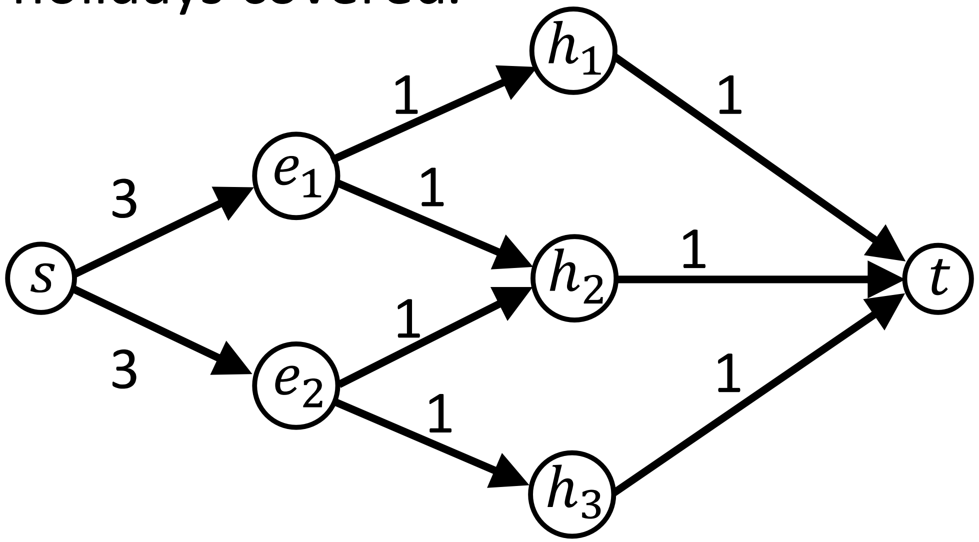
Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1.

Work Scheduling

<u>Problem:</u> We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

<u>Algorithm:</u>

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



<u>Optimality:</u>

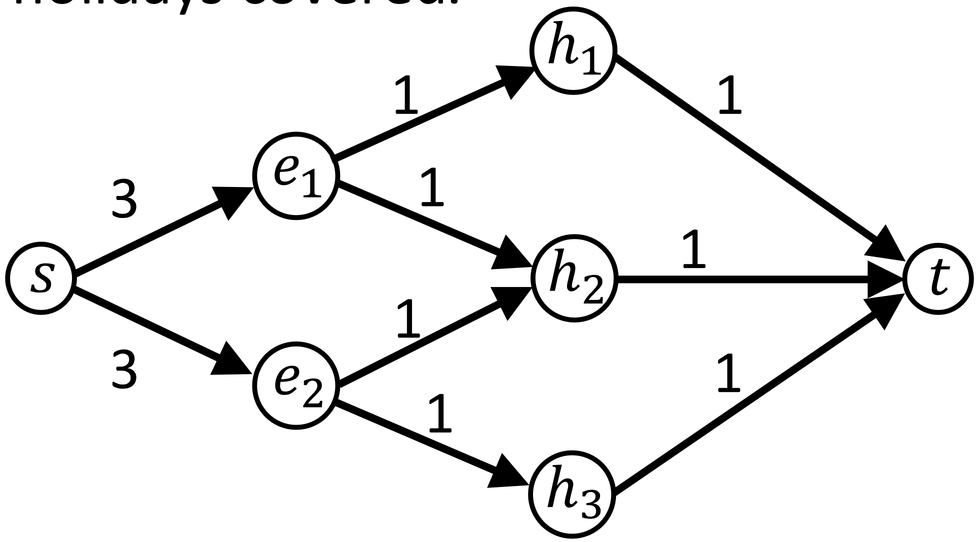
Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1. Make edge leaving each covered holiday carry 1.

Work Scheduling

<u>Problem:</u> We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

<u>Algorithm:</u>

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



<u>Optimality:</u>

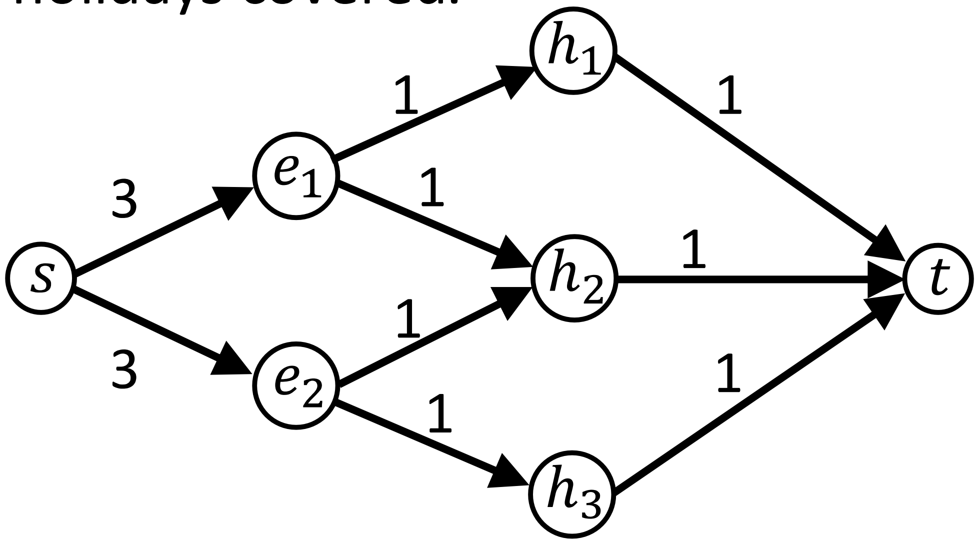
Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1. Make edge leaving each covered holiday carry 1. Make edge into each employee carry sum of the flow leaving.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

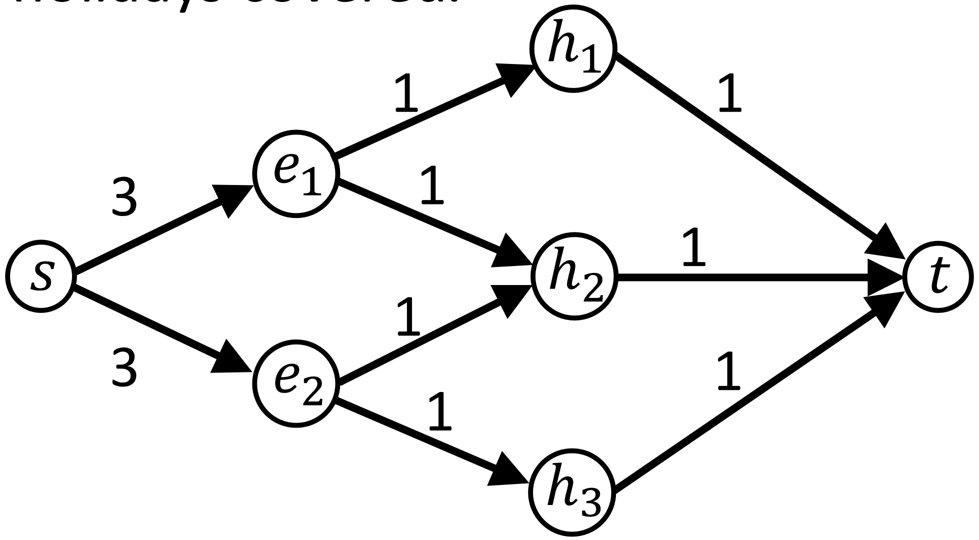


Optimality:
Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1. Make edge leaving each covered holiday carry 1. Make edge into each employee carry sum of the flow leaving.

Now we need to argue this flow is valid.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.



Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

Optimality:

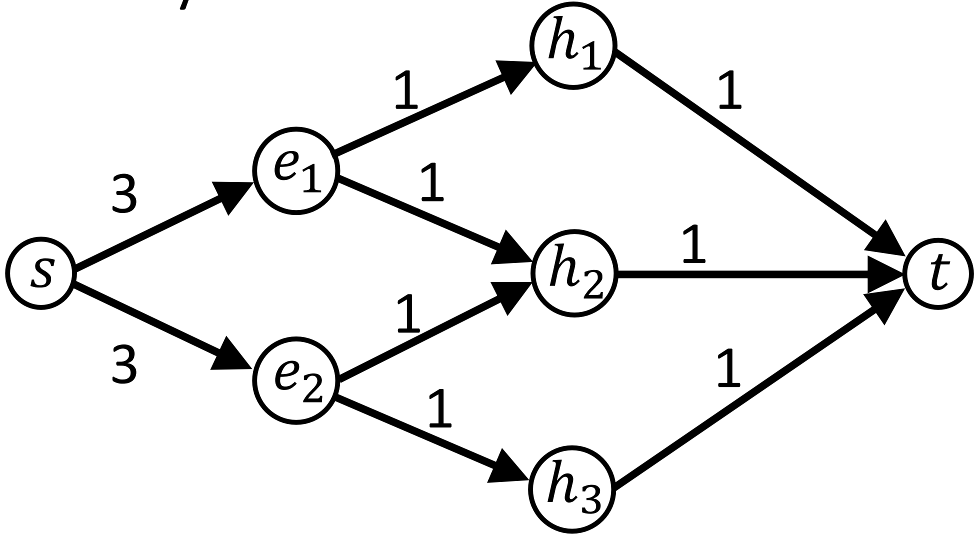
Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1. Make edge leaving each covered holiday carry 1. Make edge into each employee carry sum of the flow leaving. Conservation of flow holds at employees because…

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



Optimality:

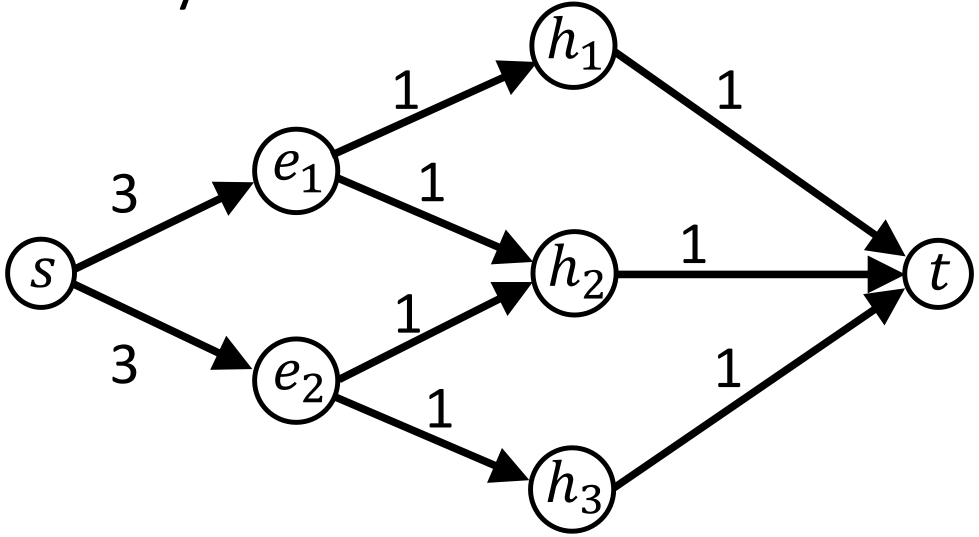
Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1. Make edge leaving each covered holiday carry 1. Make edge into each employee carry sum of the flow leaving. Conservation of flow holds at employees because that's how we set incoming flow.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

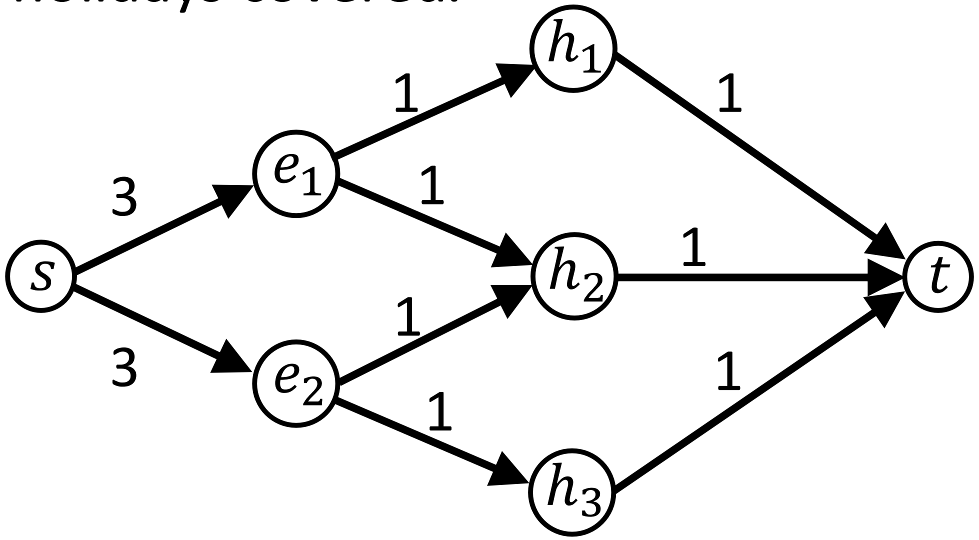


Optimality:

Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1. Make edge leaving each covered holiday carry 1. Make edge into each employee carry sum of the flow leaving. Conservation of flow holds at employees because that's how we set incoming flow. Conservation at holiday nodes holds because…

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.



Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

Optimality:

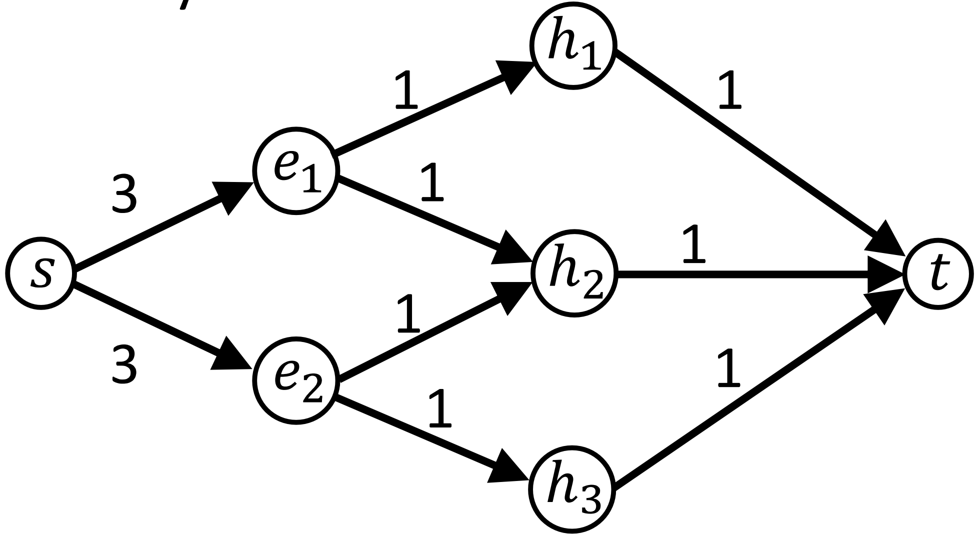
Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1. Make edge leaving each covered holiday carry 1. Make edge into each employee carry sum of the flow leaving. Conservation of flow holds at employees because that's how we set incoming flow. Conservation at holiday nodes holds because if a holiday is covered, it's done with a single employee.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

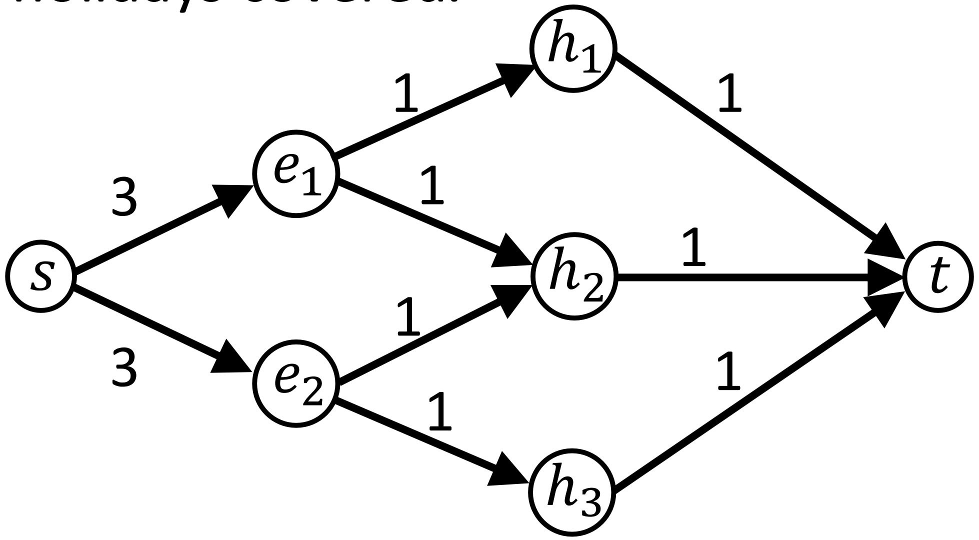


Optimality:

Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1. Make edge leaving each covered holiday carry 1. Make edge into each employee carry sum of the flow leaving. Conservation of flow holds at employees because that's how we set incoming flow. Conservation at holiday nodes holds because if a holiday is covered, it's done with a single employee. Also, no capacities are violated.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.



Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.

Optimality:

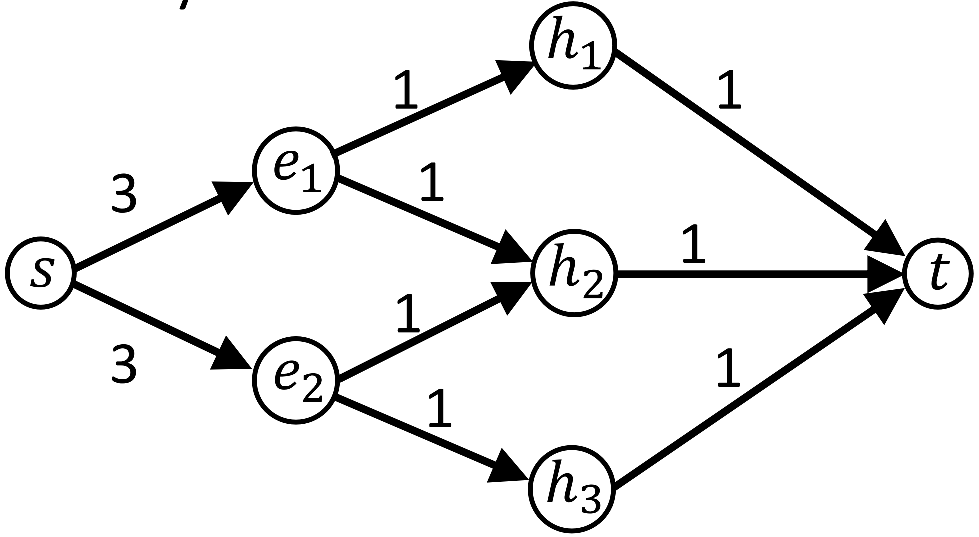
Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1. Make edge leaving each covered holiday carry 1. Make edge into each employee carry sum of the flow leaving. Conservation of flow holds at employees because that's how we set incoming flow. Conservation at holiday nodes holds because if a holiday is covered, it's done with a single employee. Also, no capacities are violated. $\Rightarrow$ Valid Flow

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



Optimality:

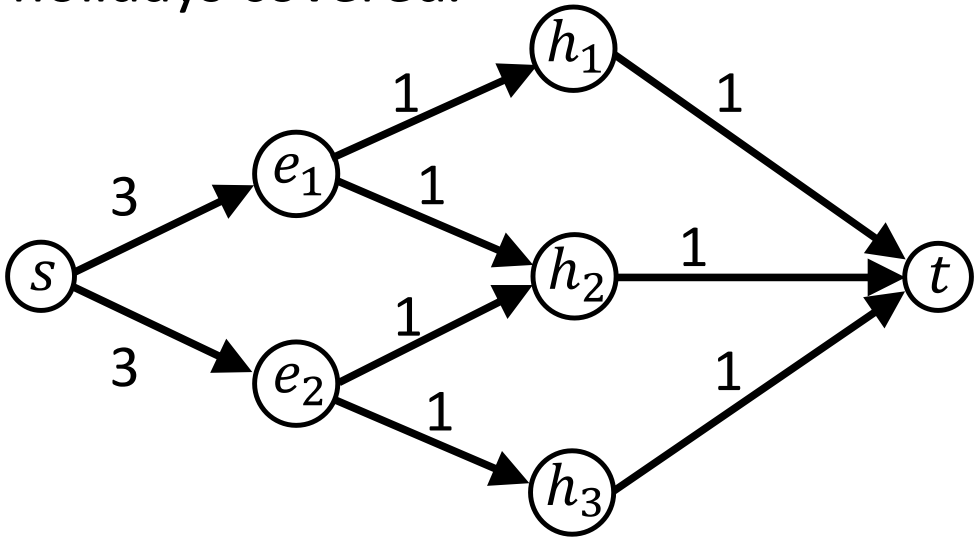
Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1. Make edge leaving each covered holiday carry 1. Make edge into each employee carry sum of the flow leaving. Conservation of flow holds at employees because that's how we set incoming flow. Conservation at holiday nodes holds because if a holiday is covered, it's done with a single employee. Also, no capacities are violated. $\Rightarrow$ Valid Flow $> F$.

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. We want to maximize the number of holidays covered.

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



Optimality:

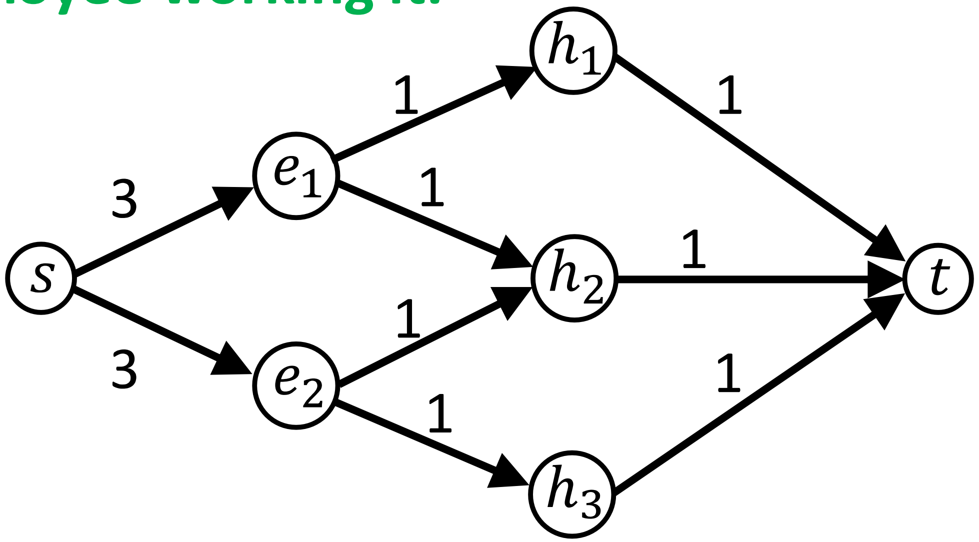
Suppose max flow $F < OPT \leq m$. That means employees can be assigned to cover OPT holidays. First, remove redundant holiday coverage. Make edge for each assignment carry 1. Make edge leaving each covered holiday carry 1. Make edge into each employee carry sum of the flow leaving. Conservation of flow holds at employees because that's how we set incoming flow. Conservation at holiday nodes holds because if a holiday is covered, it's done with a single employee. Also, no capacities are violated. $\Rightarrow$ Valid Flow $> F$. Contradiction!

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. **Each holiday needs exactly one employee working it.**

Algorithm:

1. Build flow network.
2. Find max flow.
3. If employee has outgoing edge carrying flow, assign them to work that holiday.



Need to change anything?

Work Scheduling

Problem: We need to make holiday schedules for our employees. Each employee has a set of holidays that they are able to work. Each employee should work at most 3 holidays. **Each holiday needs exactly one employee working it.**

Algorithm:

1. Build flow network.
2. Find max flow.
3. If max flow < # holidays, infeasible instance.
4. If employee has outgoing edge carrying flow, assign them to work that holiday.