Q1

Since T is tree, it is acyclic and connected. For $e \in E'$ with $e = (u, v)$, removing the edge e would result in V splitting into two disjoint sets of vertices: U and $V - U$, with $u \in U$, $v \in V - U$. Note that e has exactly one endpoint in U and exactly one endpoint in $V - U$.

Since T is the MST of G, so e is the shortest edge connecting U and $V - U$. Suppose a shorter edge e' existed that connected U and $V - U$, with $w(e') < w(e)$. Then T wouldn't be the MST since $w(E' - \{e\} \cup \{e'\}) = w(E') - w(e) + w(e') \leq w(E')$. By contradiction, e is the shortest (min cost) edge connecting U and $V - U$.

Thus, there exists a subset of vertices $S_e = U$ such that e is the min cost edge with exactly one endpoint in S_e.

Q2

(A) We first construct a DAG for the problem. We define vertices s, t, X_i for $1 \leq i \leq m$, and C_j for $1 \leq j \leq n$. The (directed) edges of the graph are:

- (s, C_j) (with weight 1) for all $1 \leq j \leq n$ where $L_j = 1$. This represents the courses that have been taken are available to be matched against the requirements.
- (C_j, X_i) (with weight 1) for all $1 \leq i \leq m$, $1 \leq j \leq n$ where $X_{i,j} = 1$. This represents the matching of the requirements.
- (X_i, t) (with weight N_i) for all $1 \leq i \leq m$. This represents the flow of the minimum number of requirements needed.

We then run the Ford-Fulkerson Max-Flow algorithm on this graph. If a flow exists where $f(s,t) = \sum N_i$ (i.e. all requirements are satisfied), then that means the student can graduate from the Department of Commuter Silence at Sham-Poobanana University.

(B) The heavy lifting is done by the Ford-Fulkerson Max-Flow algorithm, which runs in $O(|V||E|^2)$ time. The initialisation of the vertices and edges and the checking performed in the end will be considered negligible.

According to (A), we have $|V| = m+n+2$ and $|E| \leq n+mn+m$. For $|E|$, we can only be certain of the upper bound, which is when $L_j = 1 \forall j$ and $X_i, j = 1 \forall i, j$. This is the extreme case when all edges are present.

We then have

$$\begin{aligned}
O(|V||E|^2) &= O((m+n+2)(n+mn+m)) \\
&= O((m+n)(n(m+1)+m)) \\
&= O((2n)(n(m+2))) && m \leq n \\
&= O(2n^2(m+2)) \\
&= O(n^2 m)
\end{aligned}$$

Note that we made use of the knowledge that $m \leq n$. Assuming that $N_i \geq 1$, it only makes sense for $m \leq n$; otherwise there are more requirements than courses and it becomes impossible to complete all requirements.

(C) The vertex s represents the start of the flow. It is directly connected only to the courses taken, and with a weight of 1, since each course is only taken once and can only be counted once. This is the outward-flow from s, and the capacity is the number of courses taken ($\sum L_j$).

Each course C_j is linked to zero or more requirements. When running the Ford-Fulkerson Max-Flow algorithm, choosing a path with course C_j and some requirement X_i means that we attempt to register C_j to fulfill said requirement X_i. Each course can only be used to fulfill one requirement since the in-degree of each course is at most 1, so due to flow, the out-degree is also at most 1 (or at most the in-degree, to be precise).

To ensure that the requirements are satisfied, the requirement vertices X_i are directly connected to t with weight N_i. When running the algorithm, any path will have capacity 1. Selecting a path will decrement N_i in the residual graph until it reaches 0.

Finally, we perform the final check to ensure that the flow value $|f|$ equals $\sum N_i$. There may be cases where the flow value is less than $\sum N_i$, in which case this means some requirement was not satisfied. But the max flow algorithm is designed to maximise f, so if the flow value is less than $\sum N_i$, then we know there are no other flows that will satisfy the graph, and that the problem can't be solved, or in other words, the student can't graduate from the Department of Commuter Silence at Sham-Poobanana University.

Q3

(a) (i) By the Max-Flow Min-Cut algorithm in the lecture notes, if f is a maximum flow, then we can find a min-cut $c(S,T)$ for some $s-t$ cut (S,T). We first weight all edges with 1. We can determine the min-cut by running the Ford-Fulkerson Max-Flow algorithm followed by BFS on s to determine the min-cut.

(ii) Note that upon the removing the min-cut edges from G, there would be no more $s-t$ path, as required. Why? Since f is a max flow, by the min-cut theorem, the residual graph G_f has no path from s to t. The maximising property of the Ford-Fulkerson Max-Flow algorithm ensures that the edge set is minimum. Suppose it wasn't minimum, then another path could be added from s to t, and f wouldn't be a max flow.

(iii) From page 38 of the slides, running BFS uses $O(|E|)$ time. Compound this with the $|E|$ iterations needed by Ford-Fulkerson. Unlike the $O(|V||E|^2)$ implementation of Ford-Fulkerson, paths won't be visited more than once.

(b) (i) For every vertex $v \in V$, create two vertices v_l, v_r in V'. Moreover, for every vertex $v \in V - \{s,t\}$, create an edge (v_l, v_r) in E'. For every edge $(u,w) \in E$, create an edge (u_r, w_l) in E'.

(ii) If P is an $s-t$ path in G that passes through vertex v, then the corresponding path P' in G' is an $s_r - t_l$ path that passes through edge (v_l, v_r).

Q4

(a)

$$D^{(2)} = \begin{pmatrix} 0 & 5 & 10 & 11 & 3 & 5 \\ 6 & 0 & 16 & 3 & 5 & 13 \\ \infty & 3 & 0 & \infty & 1 & 3 \\ \infty & 4 & 15 & 0 & 2 & 4 \\ 8 & 2 & 7 & 3 & 0 & 2 \\ \infty & \infty & 5 & 1 & 3 & 0 \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 5 & 10 & 6 & 3 & 5 \\ 6 & 0 & 12 & 3 & 5 & 7 \\ 9 & 3 & 0 & 4 & 1 & 3 \\ 10 & 4 & 9 & 0 & 2 & 4 \\ 8 & 2 & 7 & 3 & 0 & 2 \\ 11 & 5 & 5 & 1 & 3 & 0 \end{pmatrix} = D^{(8)}$$

(b)

$$D^{(1)} = \begin{pmatrix} 0 & 8 & 10 & \infty & 3 & \infty \\ 6 & 0 & 16 & 3 & 9 & \infty \\ \infty & \infty & 0 & \infty & 1 & \infty \\ \infty & \infty & \infty & 0 & 2 & 10 \\ \infty & 2 & \infty & \infty & 0 & 2 \\ \infty & \infty & 5 & 1 & \infty & 0 \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & 8 & 10 & 11 & 3 & \infty \\ 6 & 0 & 16 & 3 & 9 & \infty \\ \infty & \infty & 0 & \infty & 1 & \infty \\ \infty & \infty & \infty & 0 & 2 & 10 \\ 8 & 2 & 18 & 5 & 0 & 2 \\ \infty & \infty & 5 & 1 & \infty & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 8 & 10 & 11 & 3 & \infty \\ 6 & 0 & 16 & 3 & 9 & \infty \\ \infty & \infty & 0 & \infty & 1 & \infty \\ \infty & \infty & \infty & 0 & 2 & 10 \\ 8 & 2 & 18 & 5 & 0 & 2 \\ \infty & \infty & 5 & 1 & 6 & 0 \end{pmatrix}$$

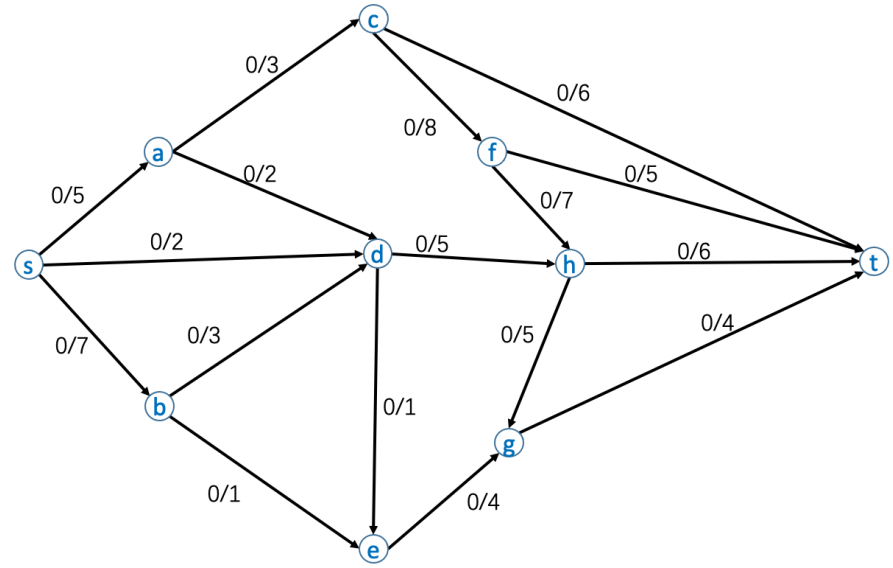
$$D^{(4)} = \begin{pmatrix} 0 & 8 & 10 & 11 & 3 & 21 \\ 6 & 0 & 16 & 3 & 5 & 13 \\ \infty & \infty & 0 & \infty & 1 & \infty \\ \infty & \infty & \infty & 0 & 2 & 10 \\ 8 & 2 & 18 & 5 & 0 & 2 \\ \infty & \infty & 5 & 1 & 3 & 0 \end{pmatrix}$$

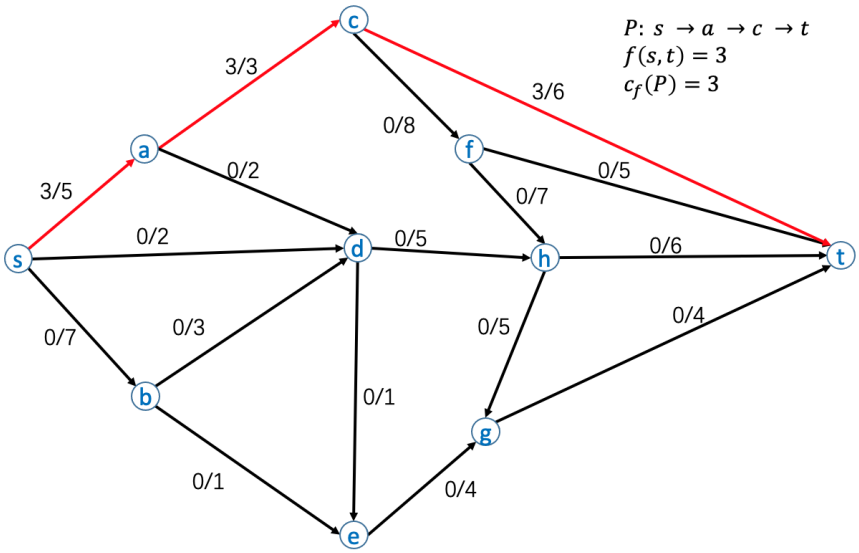
$$D^{(5)} = \begin{pmatrix} 0 & 5 & 10 & 8 & 3 & 5 \\ 6 & 0 & 16 & 3 & 5 & 7 \\ 9 & 3 & 0 & 6 & 1 & 3 \\ 10 & 4 & 20 & 0 & 2 & 4 \\ 8 & 2 & 18 & 5 & 0 & 2 \\ 11 & 5 & 5 & 1 & 3 & 0 \end{pmatrix}$$

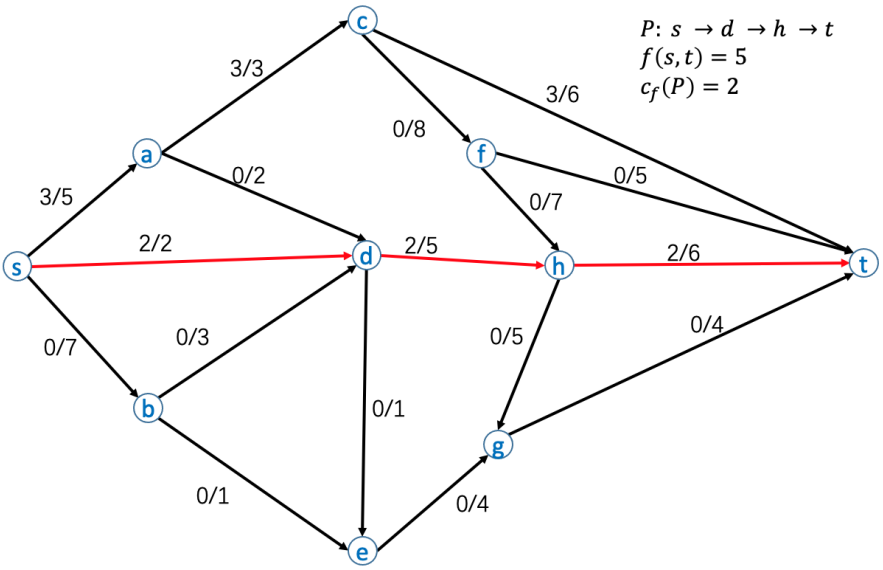
$$D^{(6)} = \begin{pmatrix} 0 & 5 & 10 & 6 & 3 & 5 \\ 6 & 0 & 12 & 3 & 5 & 7 \\ 9 & 3 & 0 & 4 & 1 & 3 \\ 10 & 4 & 9 & 0 & 2 & 4 \\ 8 & 2 & 7 & 3 & 0 & 2 \\ 11 & 5 & 5 & 1 & 3 & 0 \end{pmatrix}$$

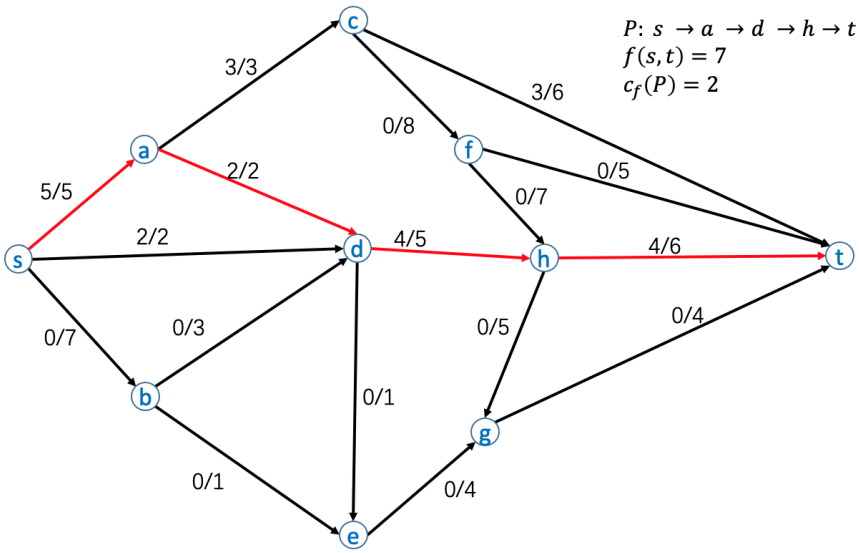
Q5

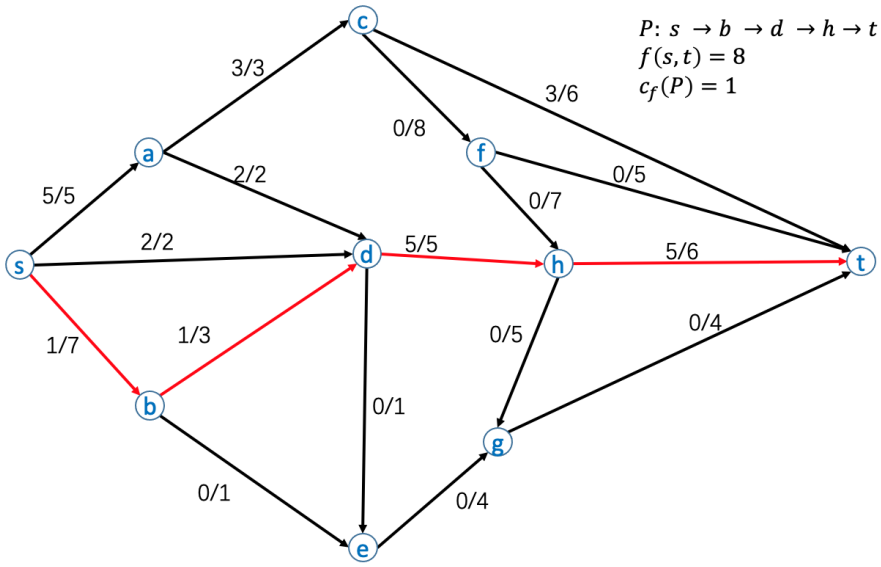
(a)

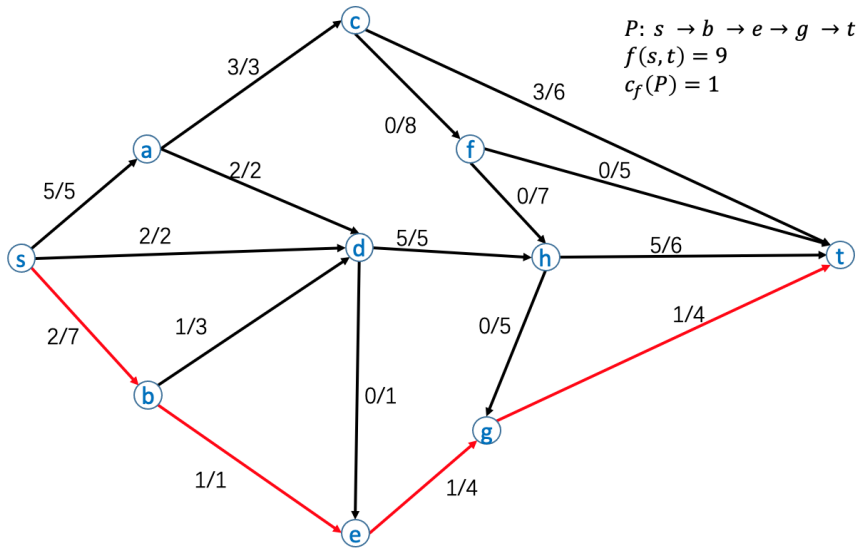

c
a
f
s
d
h
t
b
g
e
0/3
0/6
0/8
0/5
0/2
0/5
0/7
0/2
0/5
0/6
0/7
0/3
0/5
0/4
0/1
0/1
0/4

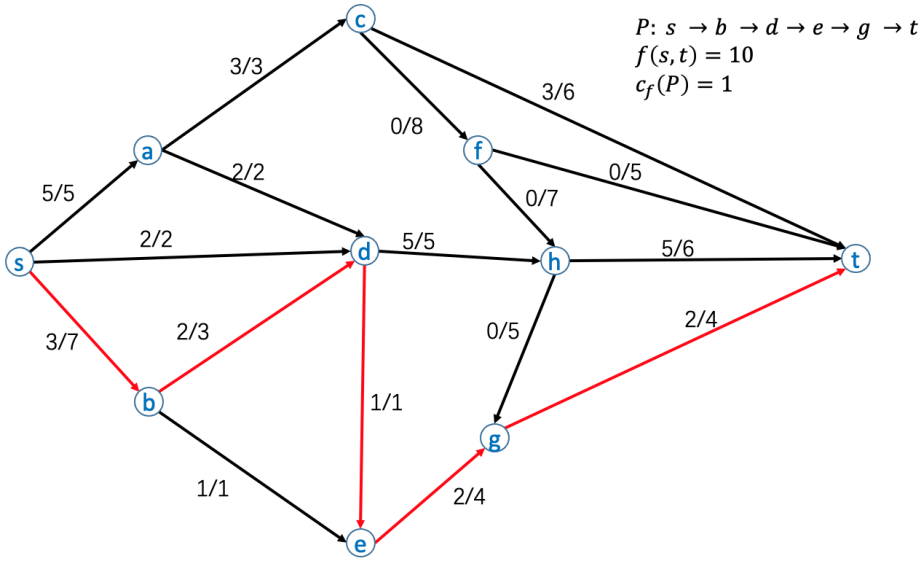

P: $s \rightarrow a \rightarrow c \rightarrow t$
$f(s,t) = 3$
$c_f(P) = 3$
c
3/3
3/6
a
f
0/8
0/2
0/5
3/5
0/7
s
0/2
d
0/5
h
0/6
t
b
g
e
0/7
0/3
0/5
0/4
0/1
0/1
0/4


P: $s \to d \to h \to t$
$f(s,t) = 5$
$c_f(P) = 2$
c
3/3
3/6
0/8
a
f
0/2
0/5
3/5
0/7
2/2
2/5
2/6
s
d
h
t
0/7
0/3
0/5
0/4
b
0/1
g
0/1
0/4
e

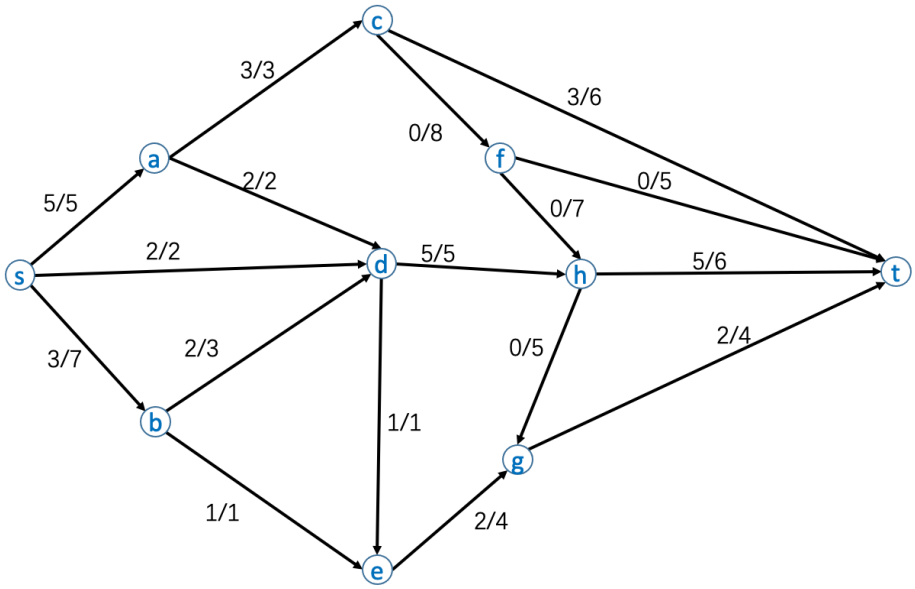

P: $s \to a \to d \to h \to t$
$f(s,t) = 7$
$c_f(P) = 2$
c
3/3
3/6
0/8
a
f
2/2
0/5
5/5
0/7
2/2
4/5
4/6
s
d
h
t
0/7
0/3
0/5
0/4
b
0/1
g
0/1
0/4
e

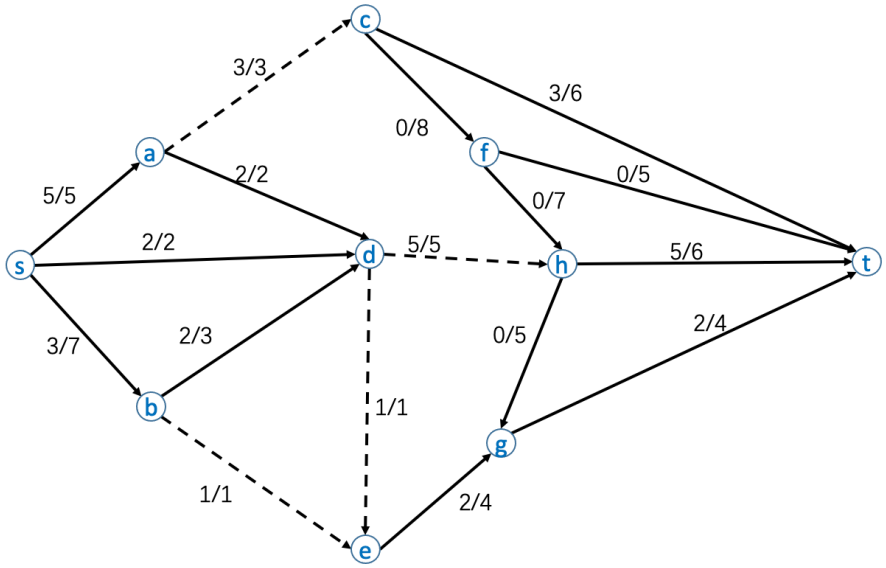

P: $s \to b \to d \to h \to t$
$f(s,t) = 8$
$c_f(P) = 1$
c
3/3
3/6
0/8
a
f
2/2
0/5
5/5
0/7
2/2
5/5
5/6
s
d
h
t
1/7
1/3
0/5
0/4
b
0/1
g
0/1
0/4
e


P: $s \rightarrow b \rightarrow e \rightarrow g \rightarrow t$
$f(s,t) = 9$
$c_f(P) = 1$
c
3/3
3/6
0/8
a
f
2/2
0/5
5/5
0/7
s
2/2
d
5/5
h
5/6
t
2/7
1/3
0/5
1/4
b
0/1
g
1/1
1/4
e


P: $s \rightarrow b \rightarrow d \rightarrow e \rightarrow g \rightarrow t$
$f(s,t) = 10$
$c_f(P) = 1$
c
3/3
3/6
0/8
a
f
2/2
0/5
5/5
0/7
s
2/2
d
5/5
h
5/6
t
3/7
2/3
0/5
2/4
b
1/1
g
1/1
2/4
e

(b)


c
3/3
3/6
0/8
a
f
2/2
0/5
5/5
0/7
2/2
s
d
5/5
h
5/6
t
3/7
2/3
0/5
2/4
b
1/1
g
1/1
2/4
e


c
3/3
3/6
0/8
a
f
2/2
0/5
5/5
0/7
2/2
s
d
5/5
h
5/6
t
3/7
2/3
0/5
2/4
b
1/1
g
1/1
2/4
e