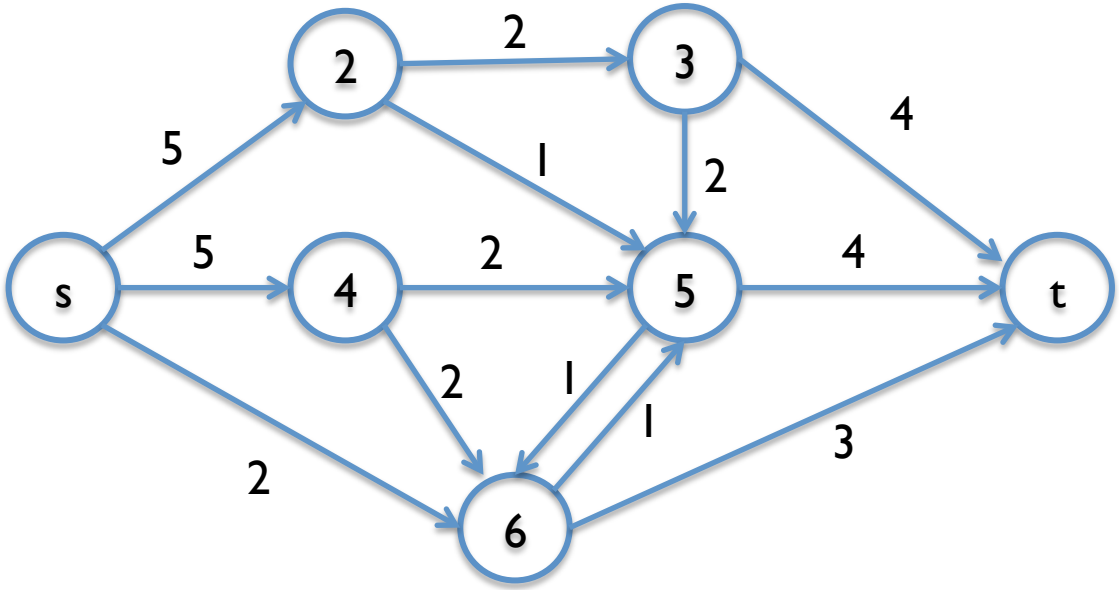
CMSC5733 Social Computing

Tutorial VII: HW3 Solution

Shenglin Zhao
The Chinese University of Hong Kong
slzhao@cse.cuhk.edu.hk

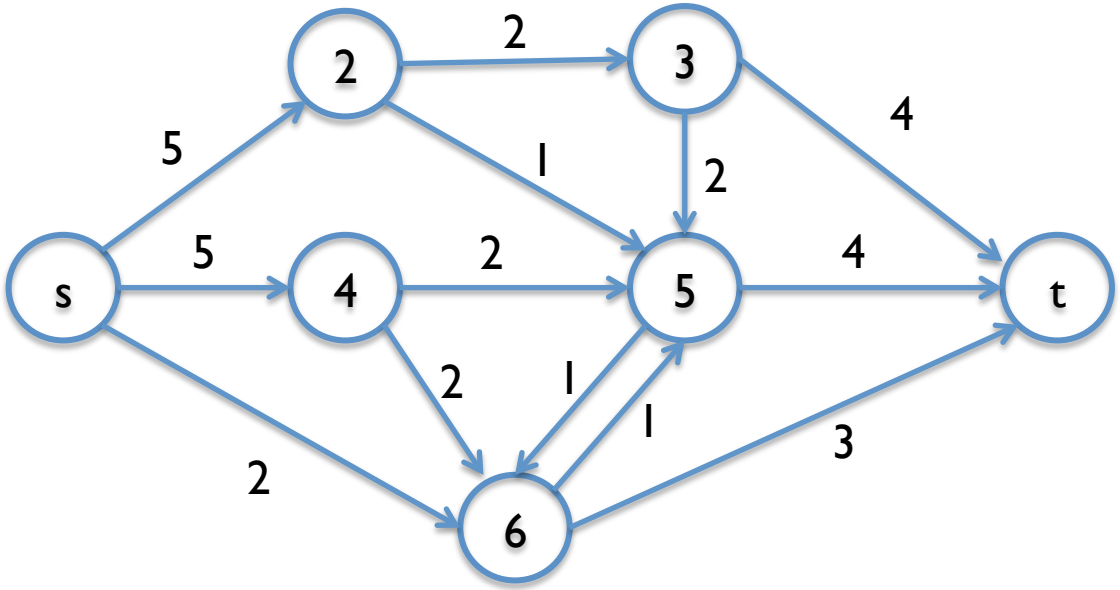
Q1

- Draw the graph

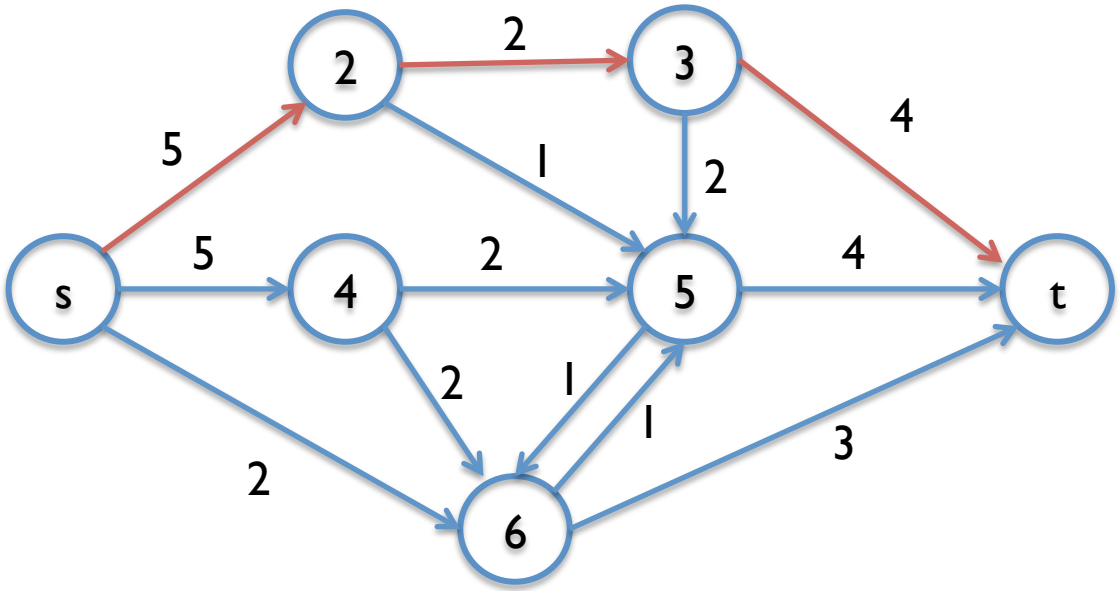


Q1

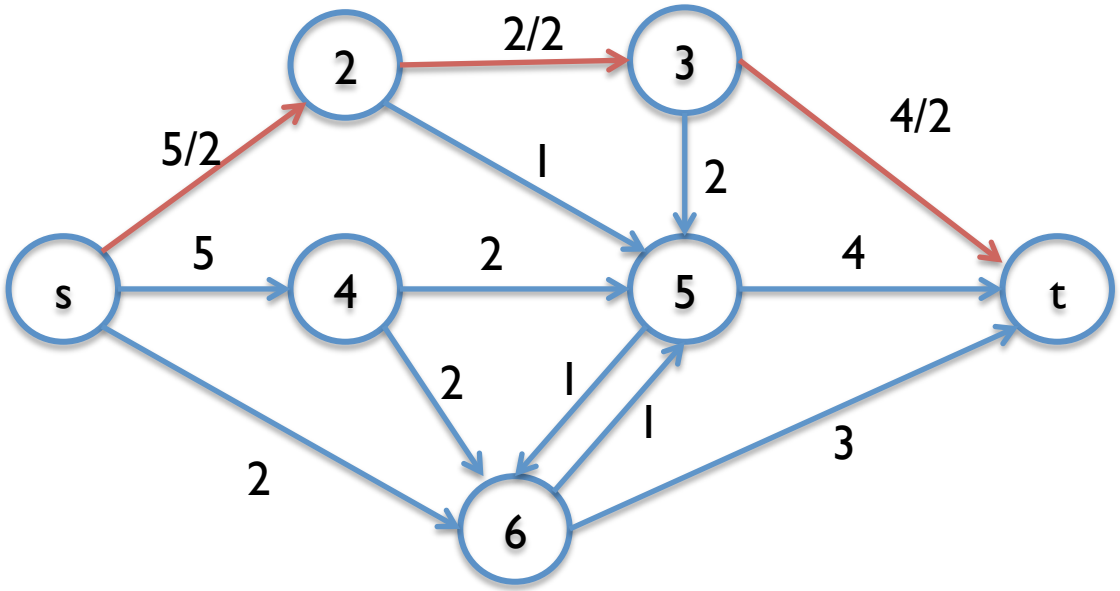
- Ford-Fulkerson algorithm



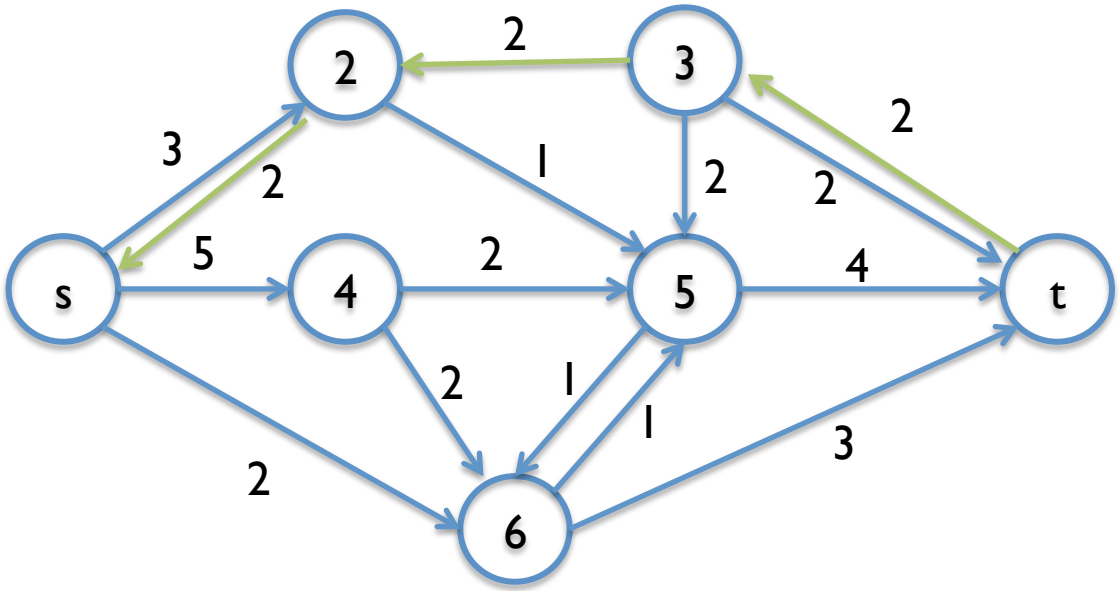
Q1



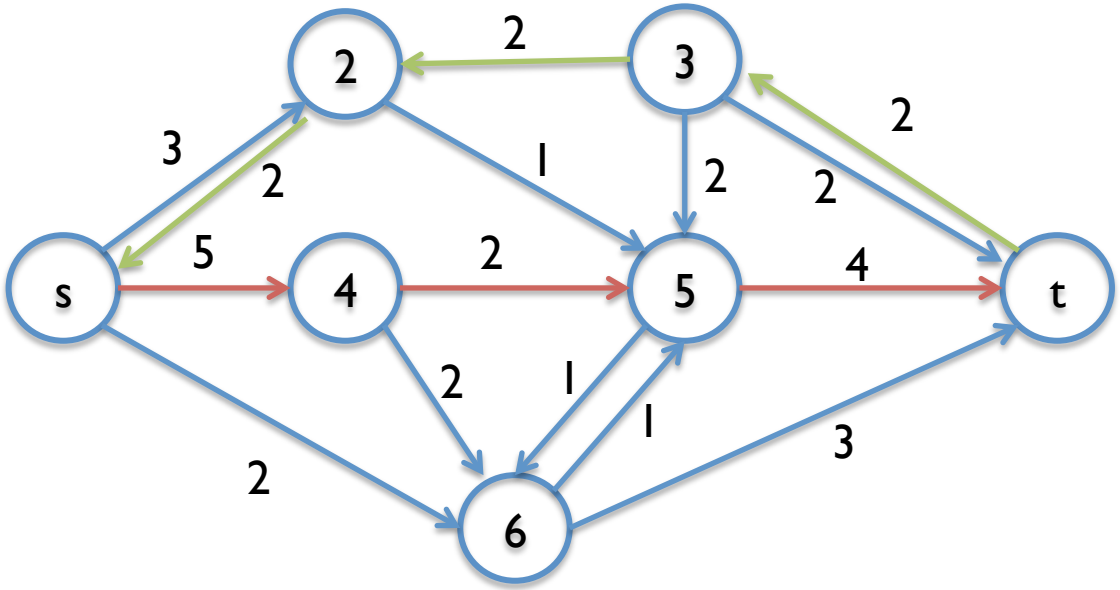
Q1



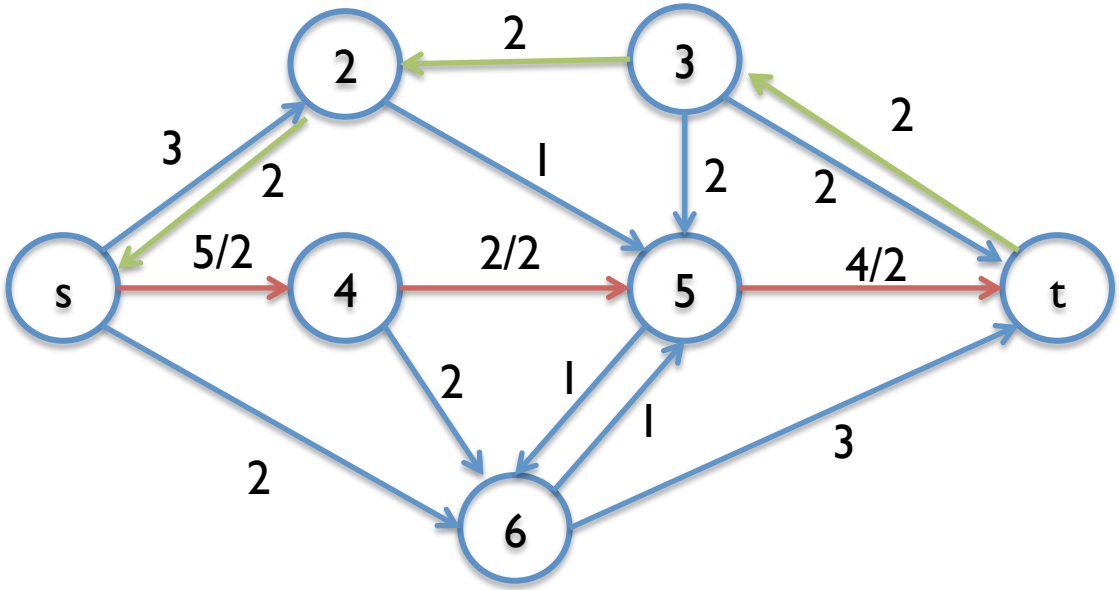
Q1



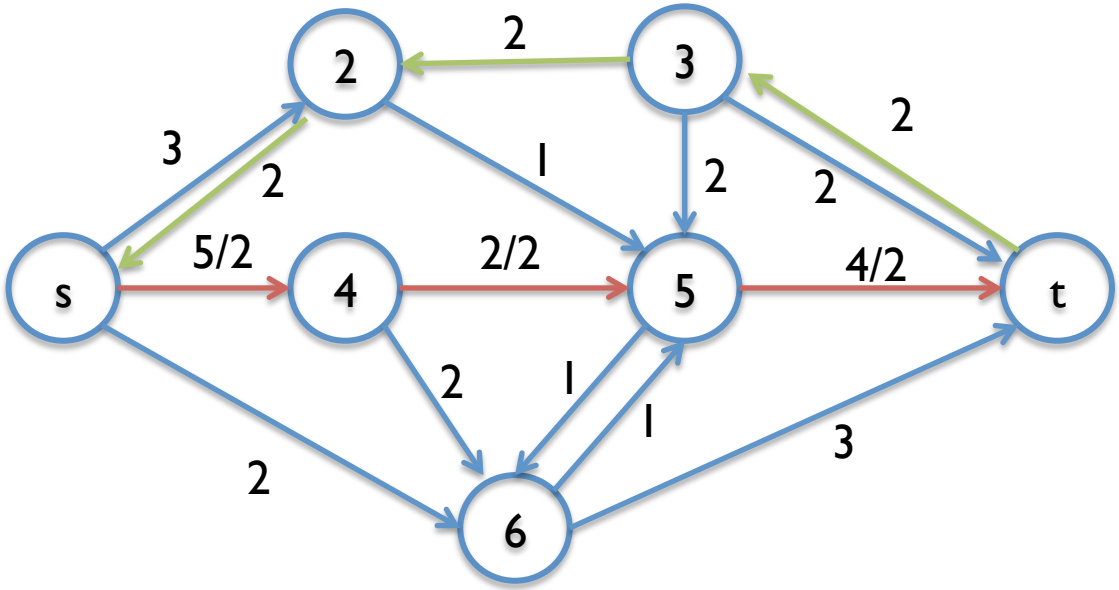
Q1



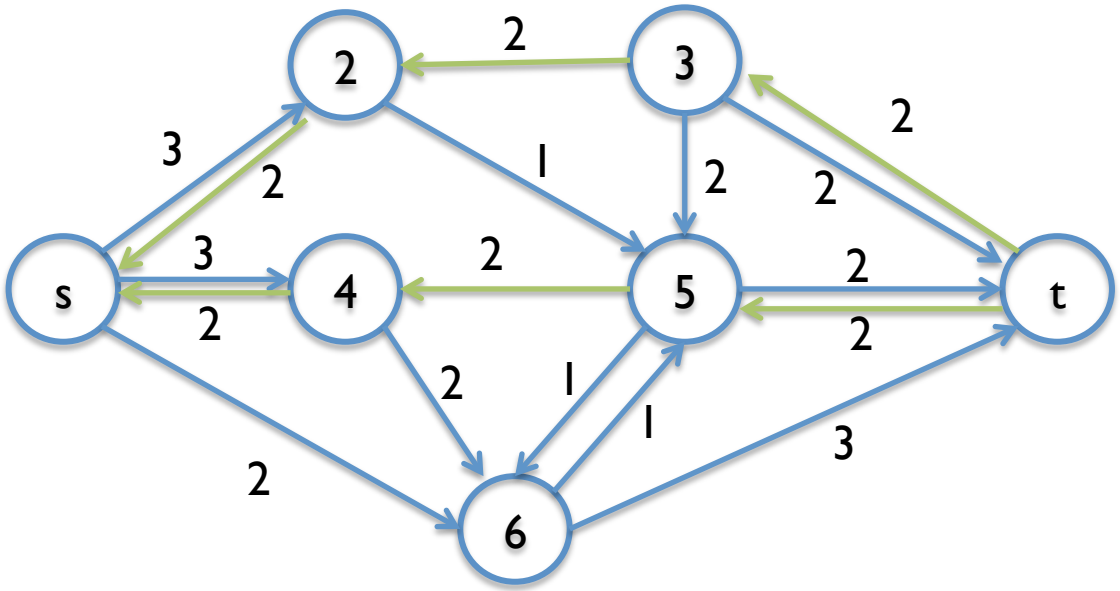
Q1



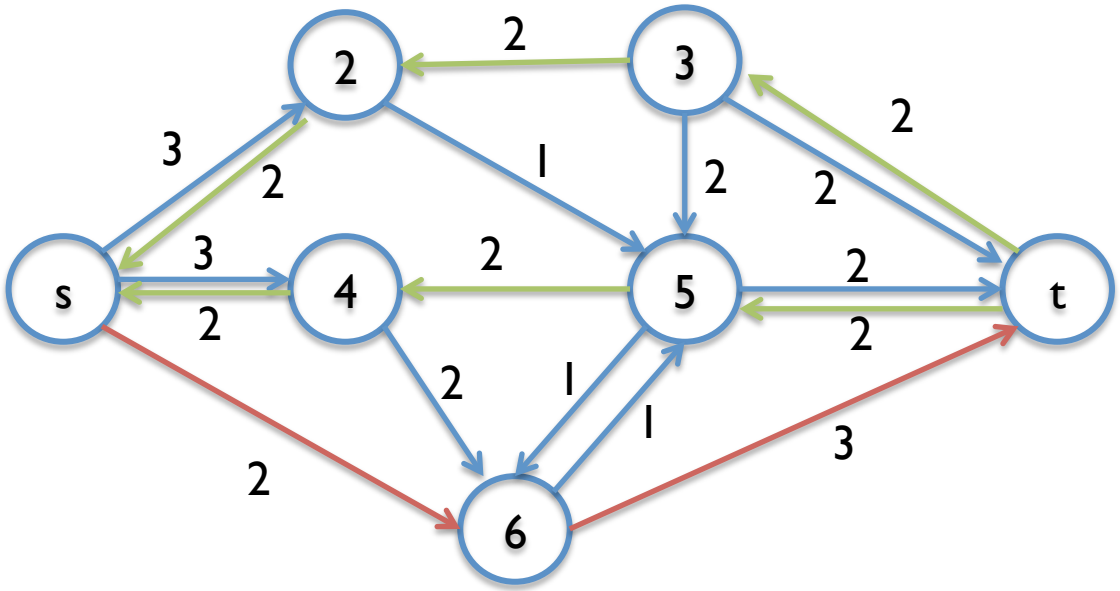
Q1



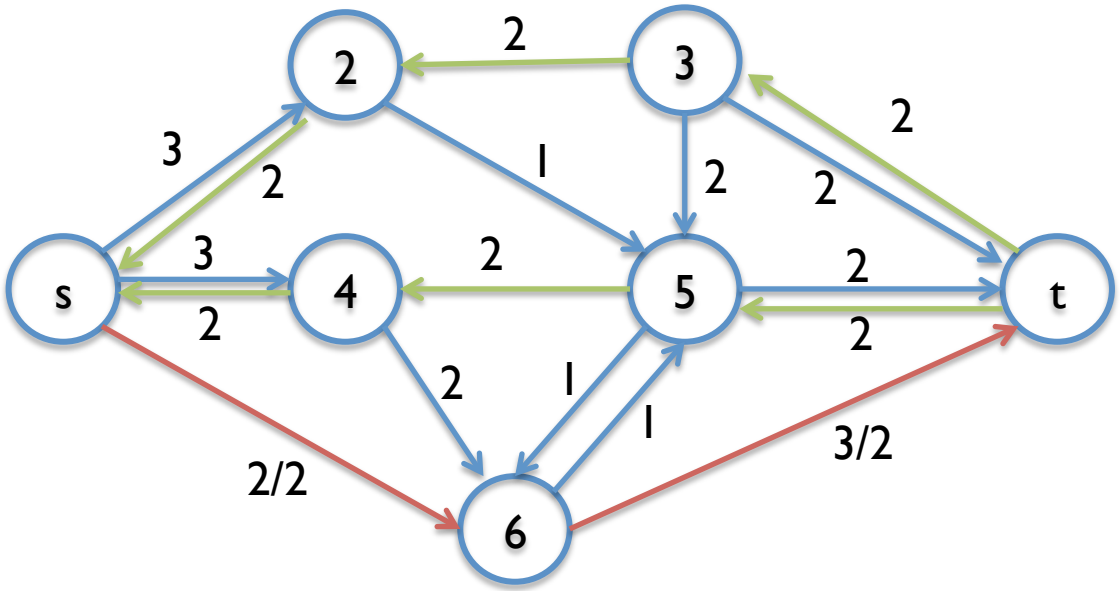
Q1



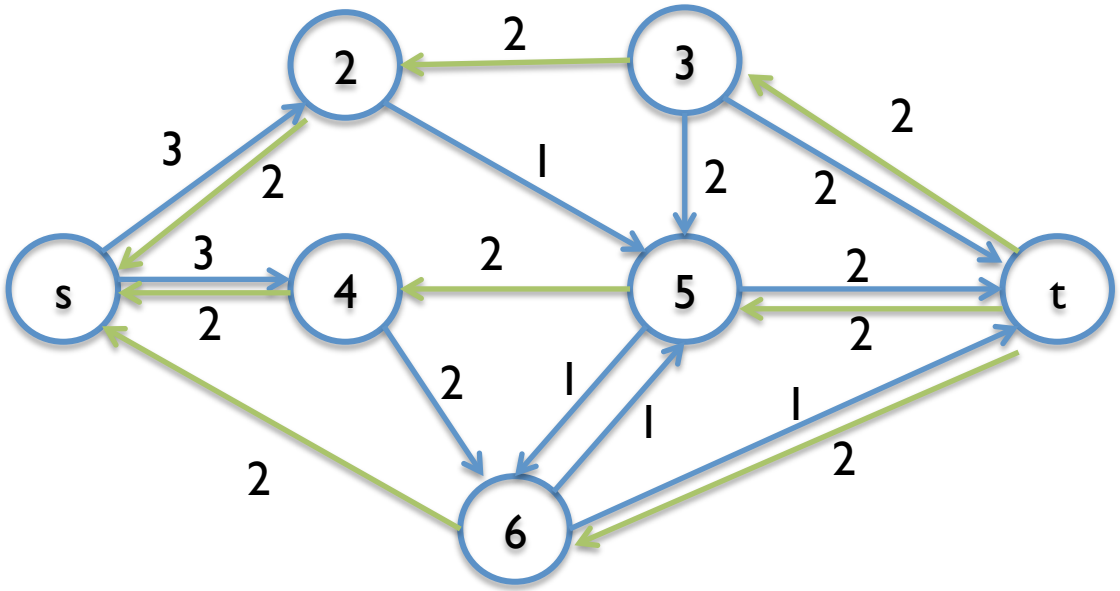
Q1



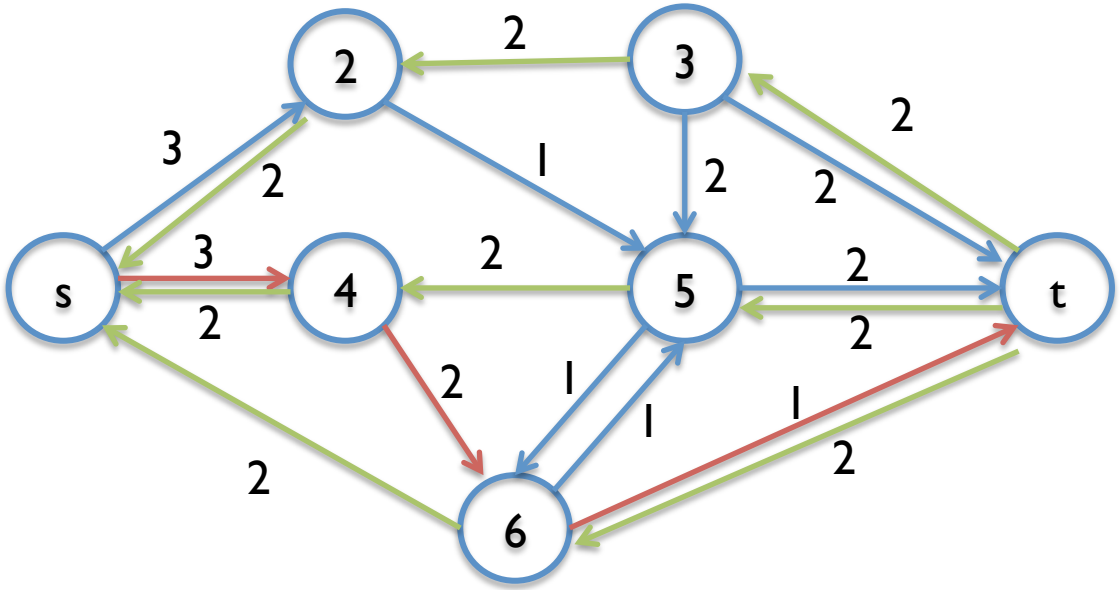
Q1



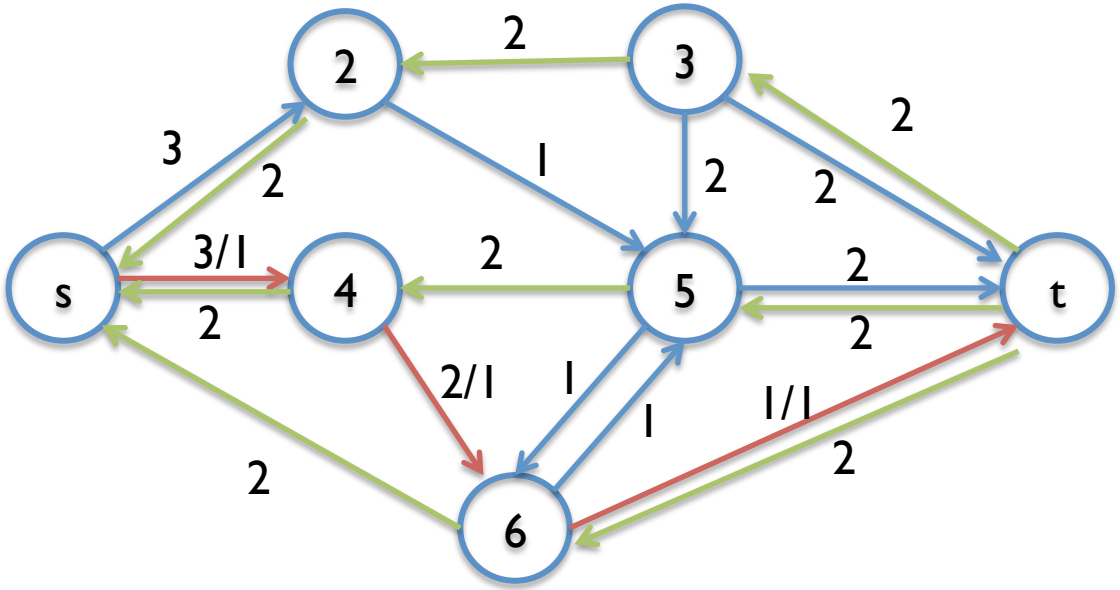
Q1



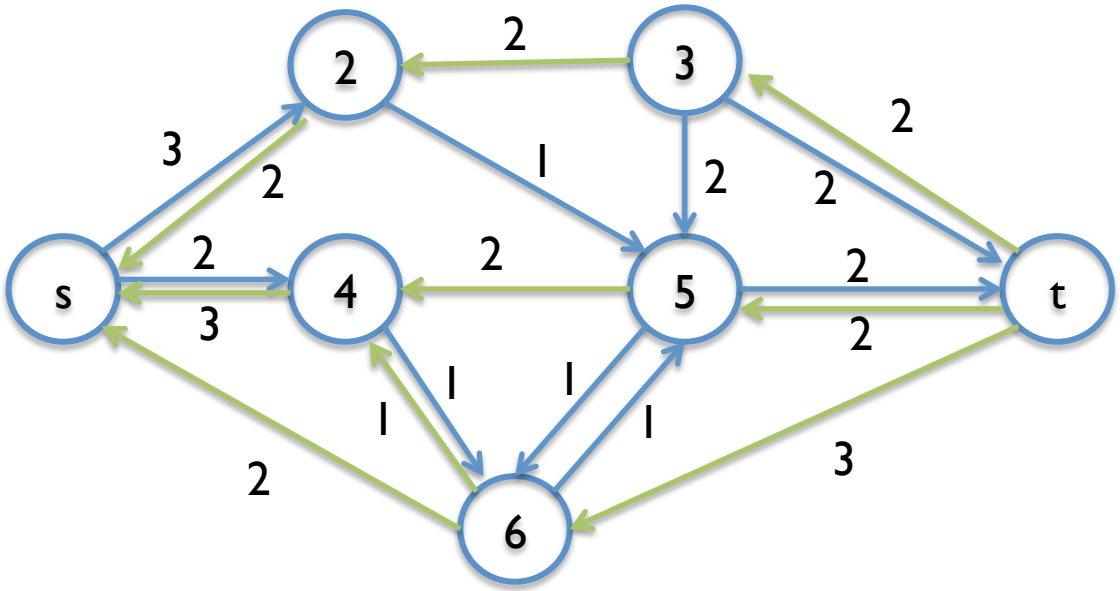
Q1



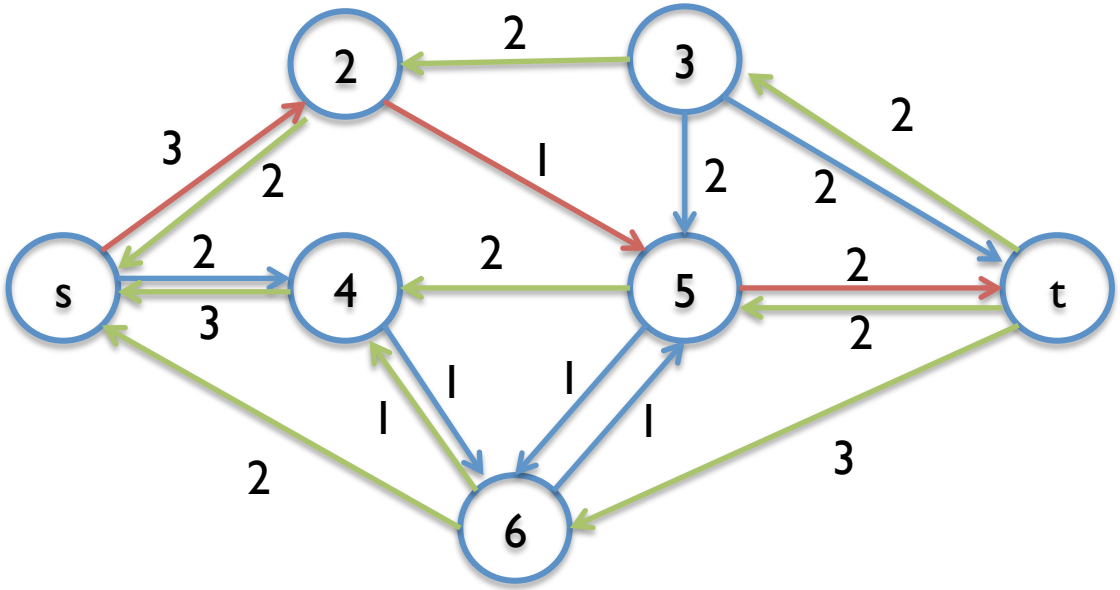
Q1



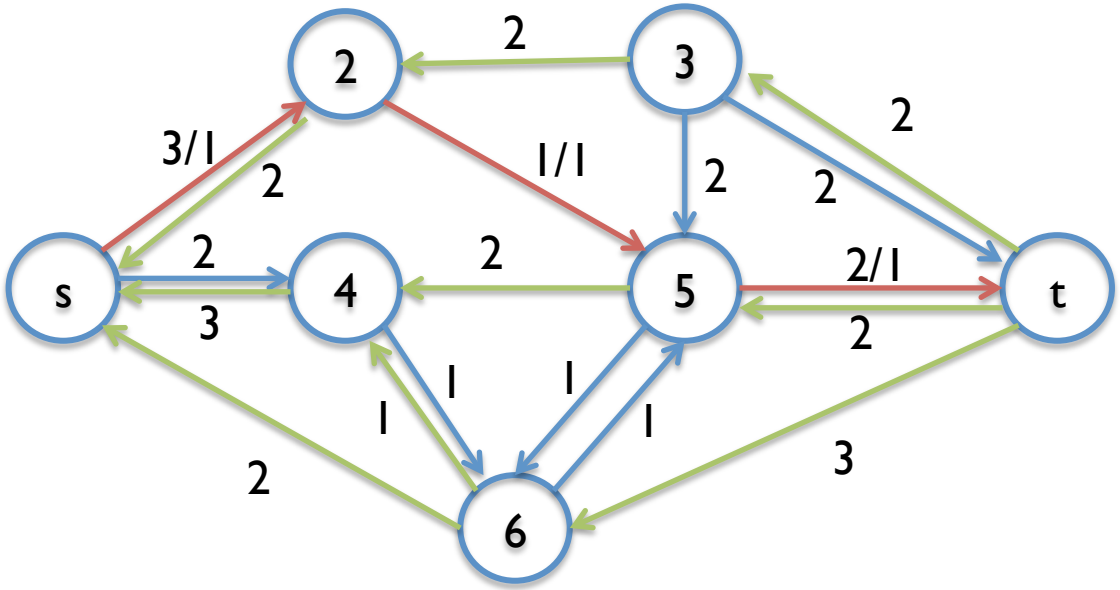
Q1



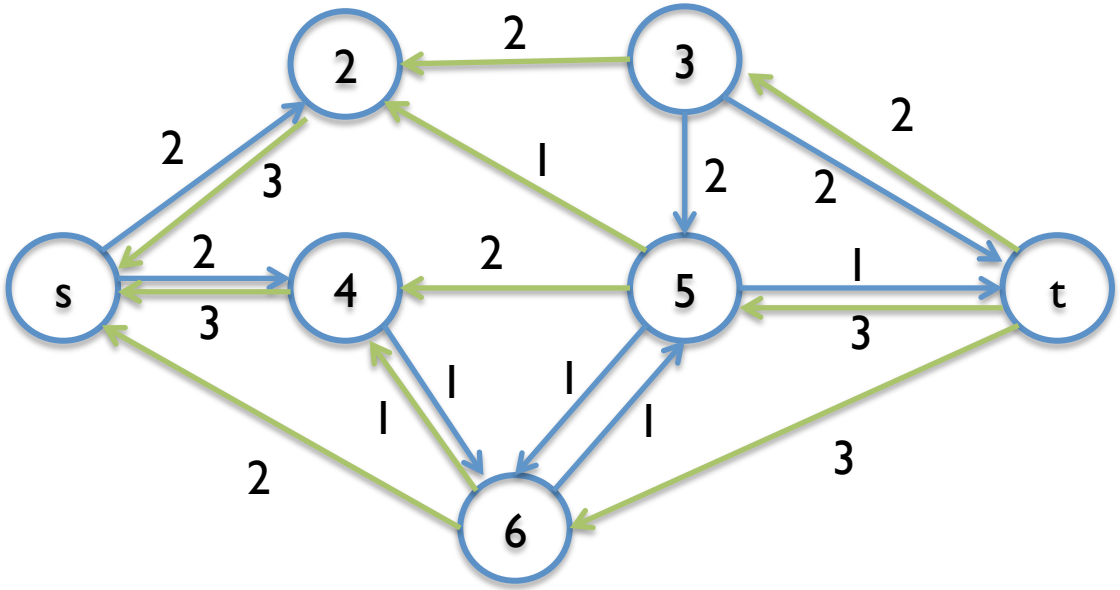
Q1



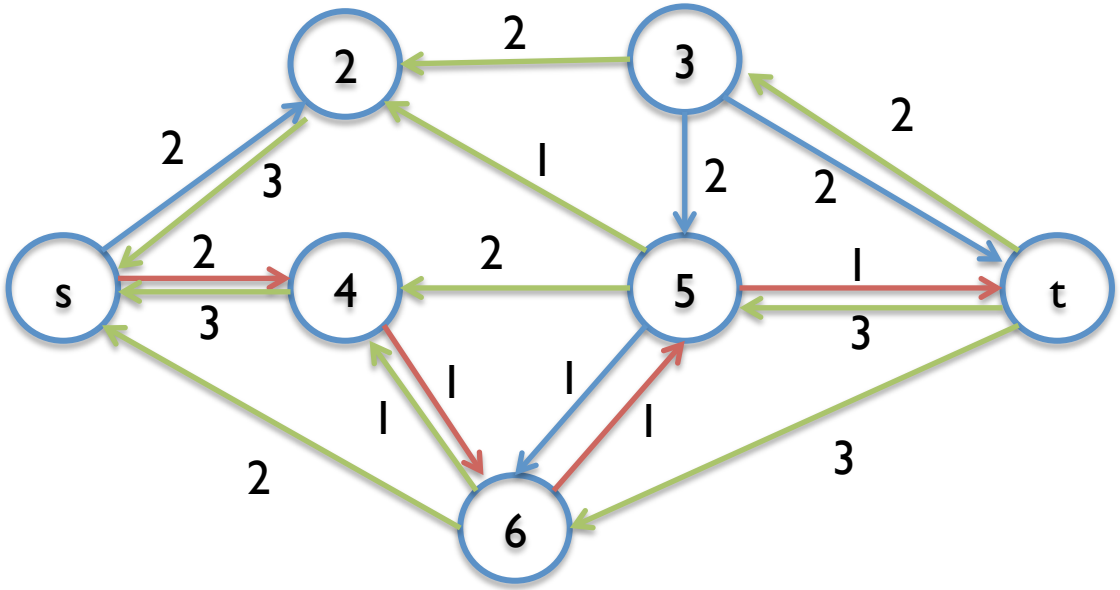
Q1



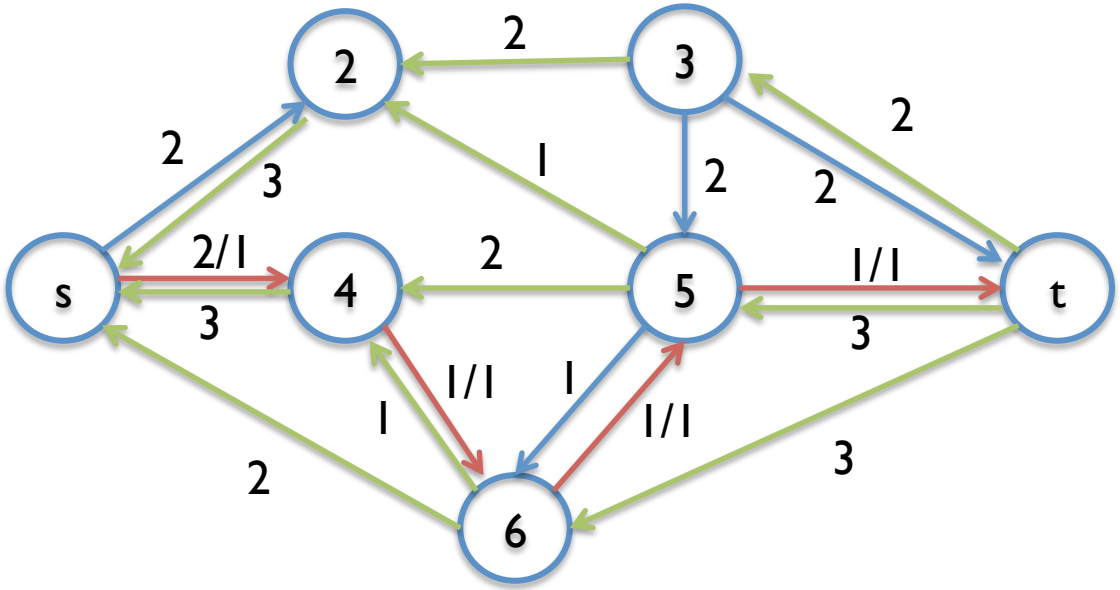
Q1



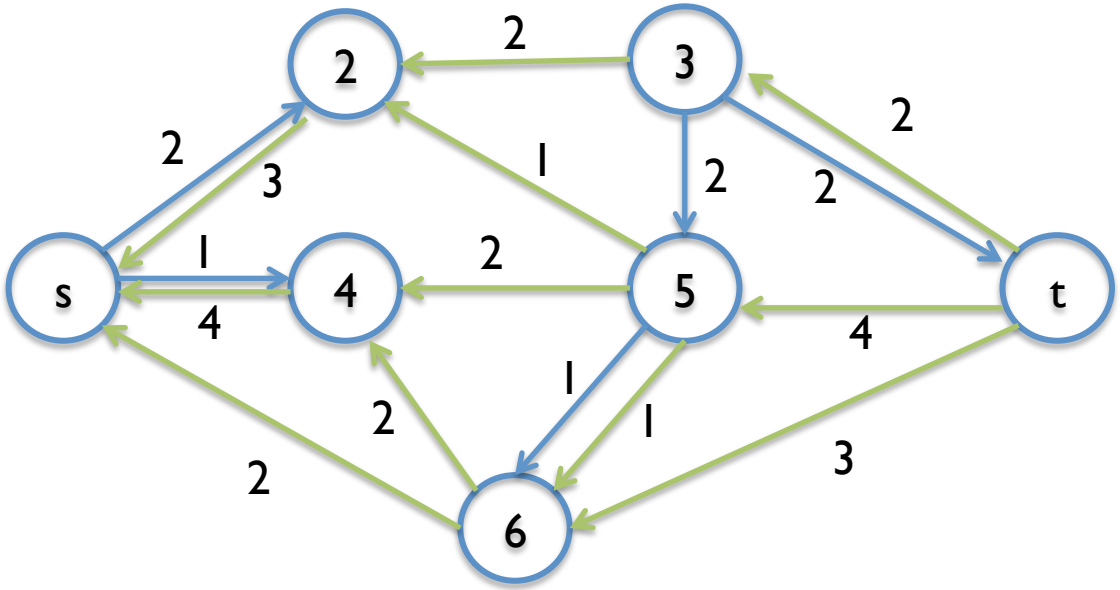
Q1



Q1

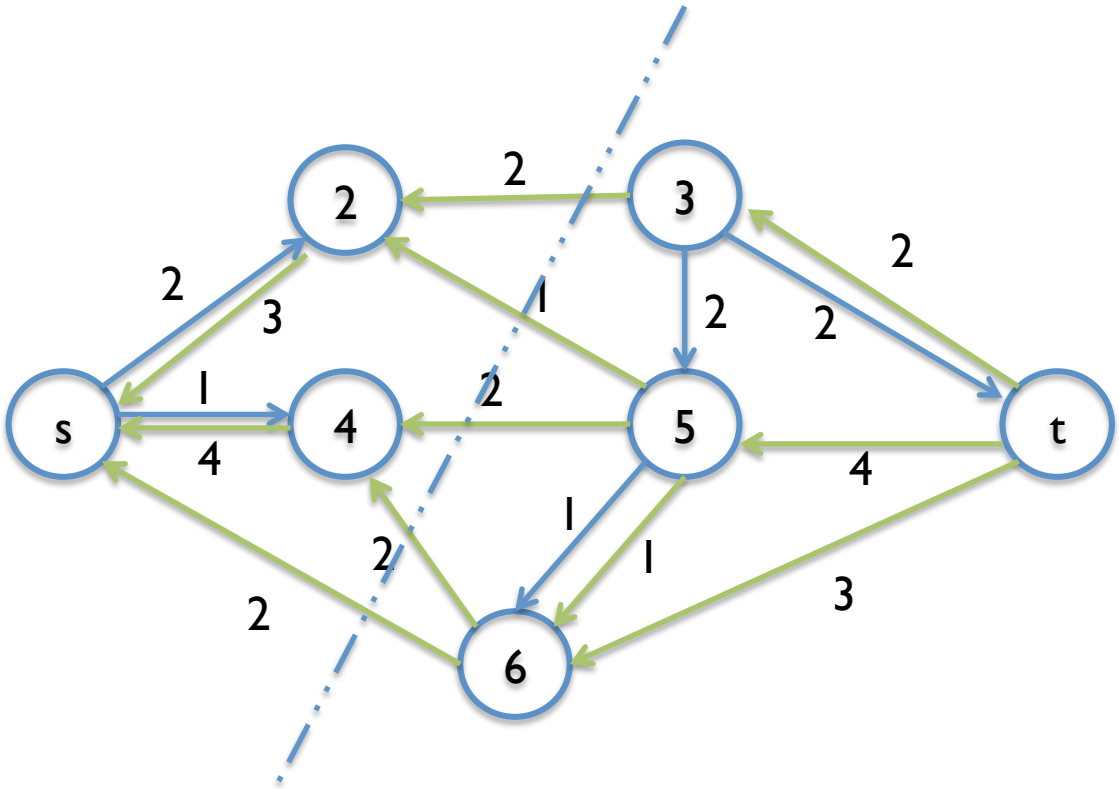


Q1



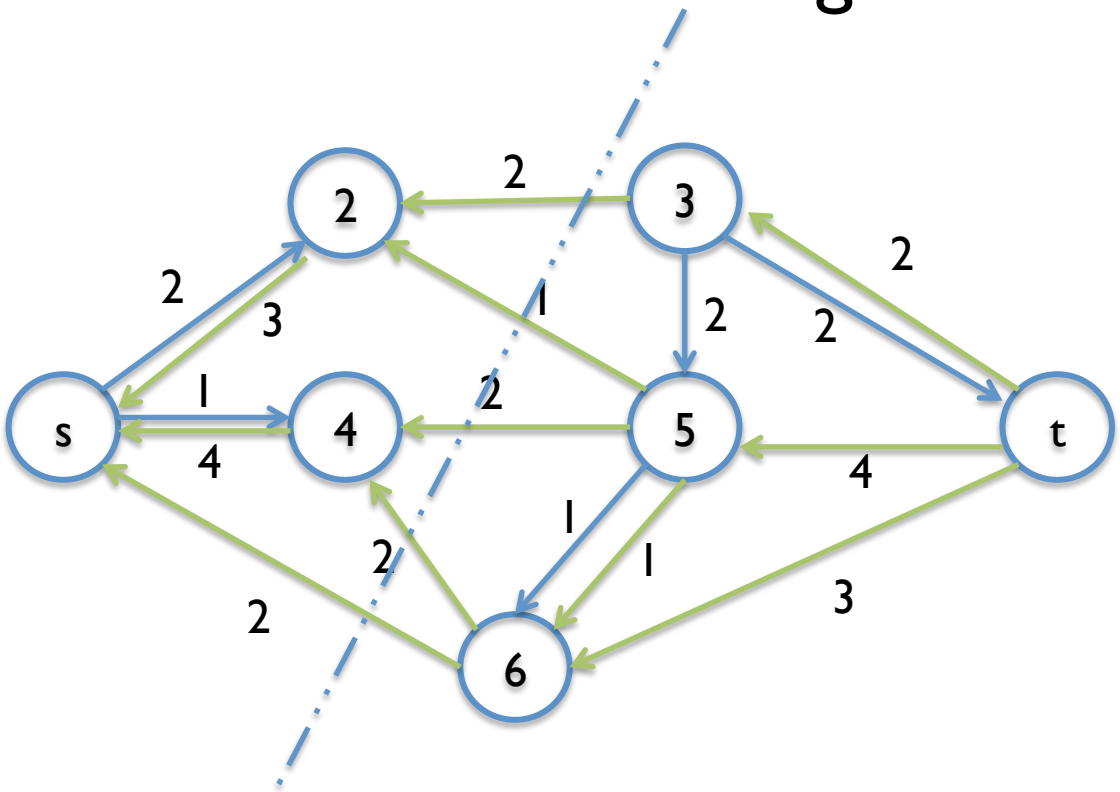
Q1

- Maximum flow is 9



Q1

- Minimum cut is the cut to get maximum flow

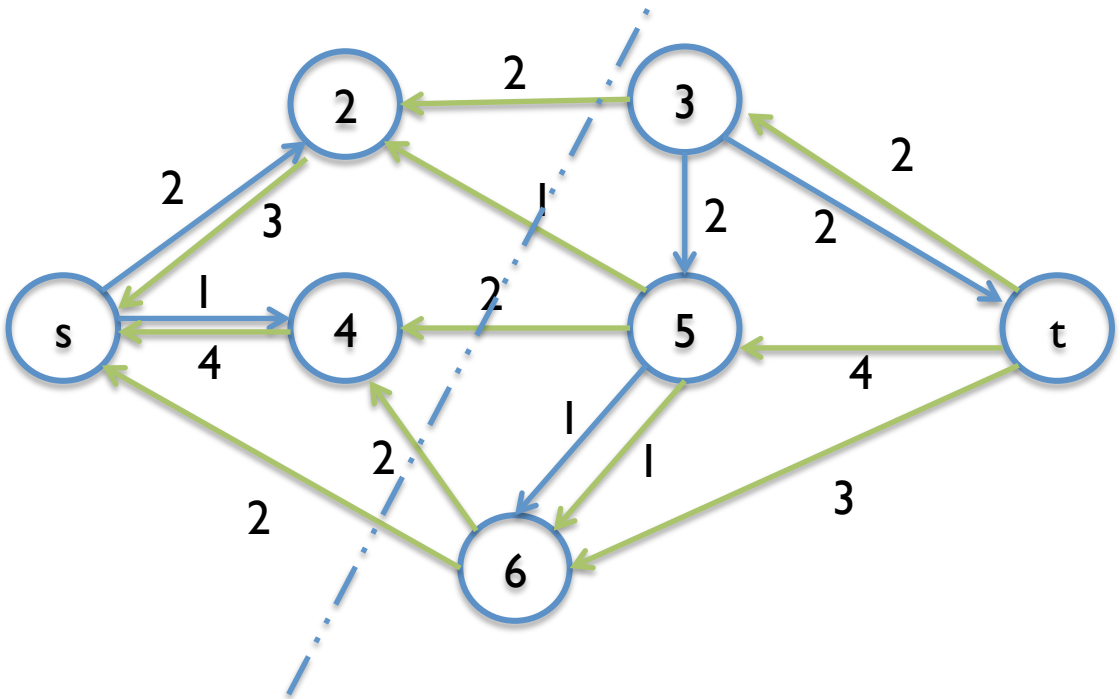


Q1

- Another way:

After the max-flow is found, the minimum cut is determined by

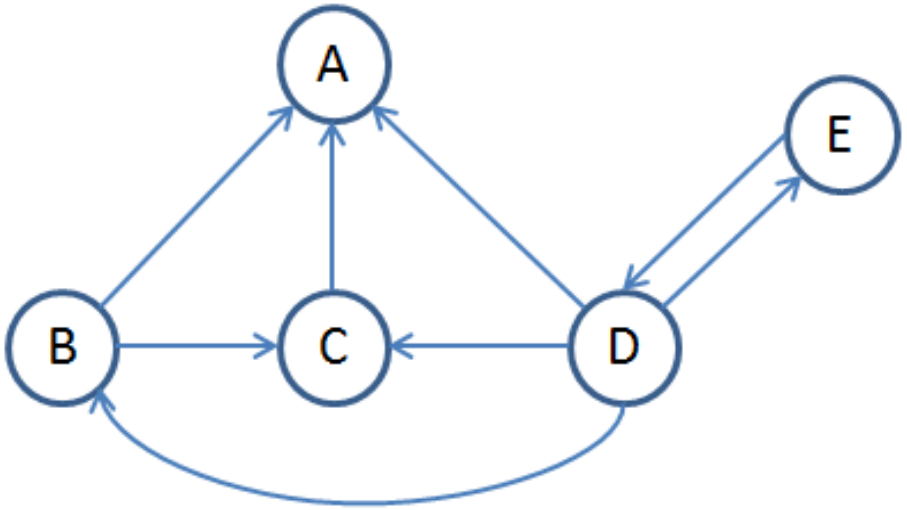
$$S = \{\text{All vertices reachable from } s\}$$
$$T = G \backslash S$$



Q2: PageRank and HITS

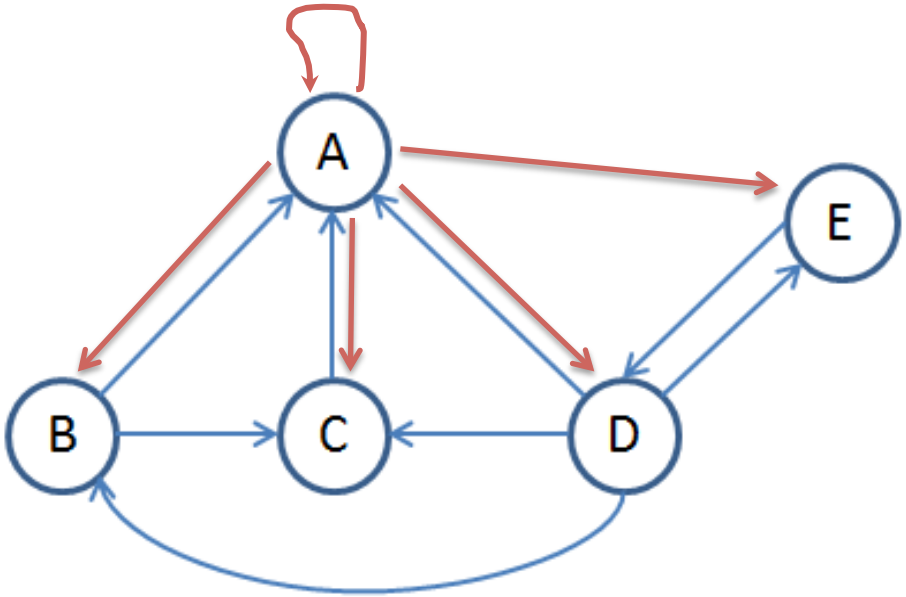
- Node A is dangling node, without outdegree
- Formula:

$$PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right)$$



PageRank and HITS

- Method to handle dangling node
  - Uniformly distribute its PageRank value to all nodes



PageRank

$$
\begin{aligned}
PR_1(A) &= \frac{1-d}{N} + d * \left(\frac{PR_0(A)}{L(A)} + \frac{PR_0(B)}{L(B)} + \frac{PR_0(C)}{L(C)} + \frac{PR_0(D)}{L(D)} \right) \\
&= \frac{0.2}{5} + 0.8 * \left(\frac{0.2}{5} + \frac{0.2}{2} + \frac{0.2}{1} + \frac{0.2}{4} \right) \\
&= 0.352
\end{aligned}
$$

$$
\begin{aligned}
PR_1(B) &= \frac{1-d}{N} + d * \left(\frac{PR_0(A)}{L(A)} + \frac{PR_0(D)}{L(D)} \right) \\
&= \frac{0.2}{5} + 0.8 * \left(\frac{0.2}{5} + \frac{0.2}{4} \right) \\
&= 0.112
\end{aligned}
$$

PageRank

$$
\begin{aligned}
PR_1(C) &= \frac{1-d}{N} + d * \left(\frac{PR_0(A)}{L(A)} + \frac{PR_0(B)}{L(B)} + \frac{PR_0(D)}{L(D)} \right) \\
&= \frac{0.2}{5} + 0.8 * \left(\frac{0.2}{5} + \frac{0.2}{2} + \frac{0.2}{4} \right) \\
&= 0.192
\end{aligned}
$$

$$
\begin{aligned}
PR_1(D) &= \frac{1-d}{N} + d * \left(\frac{PR_0(A)}{L(A)} + \frac{PR_0(E)}{L(E)} \right) \\
&= \frac{0.2}{5} + 0.8 * \left(\frac{0.2}{5} + \frac{0.2}{1} \right) \\
&= 0.232
\end{aligned}
$$

$$
\begin{aligned}
PR_1(E) &= \frac{1-d}{N} + d * \left(\frac{PR_0(A)}{L(A)} + \frac{PR_0(D)}{L(D)} \right) \\
&= \frac{0.2}{5} + 0.8 * \left(\frac{0.2}{5} + \frac{0.2}{4} \right) \\
&= 0.112
\end{aligned}
$$

PageRank

$$
\begin{aligned}
PR_2(A) &= \frac{1-d}{N} + d * \left(\frac{PR_1(A)}{L(A)} + \frac{PR_1(B)}{L(B)} + \frac{PR_1(C)}{L(C)} + \frac{PR_1(D)}{L(D)} \right) \\
&= \frac{0.2}{5} + 0.8 * \left(\frac{0.352}{5} + \frac{0.112}{2} + \frac{0.192}{1} + \frac{0.232}{4} \right) \\
&= 0.341
\end{aligned}
$$

$$
\begin{aligned}
PR_2(B) &= \frac{1-d}{N} + d * \left(\frac{PR_1(A)}{L(A)} + \frac{PR_1(D)}{L(D)} \right) \\
&= \frac{0.2}{5} + 0.8 * \left(\frac{0.352}{5} + \frac{0.232}{4} \right) \\
&= 0.143
\end{aligned}
$$

PageRank

$$
\begin{aligned}
PR_2(C) &= \frac{1-d}{N} + d * \left(\frac{PR_1(A)}{L(A)} + \frac{PR_1(B)}{L(B)} + \frac{PR_0(D)}{L(D)} \right) \\
&= \frac{0.2}{5} + 0.8 * \left(\frac{0.352}{5} + \frac{0.112}{2} + \frac{0.232}{4} \right) \\
&= 0.188
\end{aligned}
$$

$$
\begin{aligned}
PR_2(D) &= \frac{1-d}{N} + d * \left(\frac{PR_1(A)}{L(A)} + \frac{PR_1(E)}{L(E)} \right) \\
&= \frac{0.2}{5} + 0.8 * \left(\frac{0.352}{5} + \frac{0.112}{1} \right) \\
&= 0.186
\end{aligned}
$$

$$
\begin{aligned}
PR_2(E) &= \frac{1-d}{N} + d * \left(\frac{PR_1(A)}{L(A)} + \frac{PR_1(D)}{L(D)} \right) \\
&= \frac{0.2}{5} + 0.8 * \left(\frac{0.352}{5} + \frac{0.232}{4} \right) \\
&= 0.143
\end{aligned}
$$

HITS

- Start with each node with initial hub scores and authority scores
- **Authority Update**: Update each node's *Authority score* to be equal to the sum of the *Hub Scores* of each node that points to it.
- **Hub Update**: Update each node's *Hub Score* to be equal to the sum of the *Authority Scores* of each node that it points to.
- Normalize the values Repeat from the second step as necessary.
- Repeat from the second step as necessary

https://en.wikipedia.org/wiki/HITS_algorithm

HITS

Let x represent authority score and y represent hub score.

- Compute authority

$$x_1(A) = y_0(B) + y_0(C) + y_0(D) = 0.2 + 0.2 + 0.2 = 0.6$$
$$x_1(B) = y_0(D) = 0.2$$
$$x_1(C) = y_0(B) + y_0(D) = 0.2 + 0.2 = 0.4$$
$$x_1(D) = y_0(E) = 0.2$$
$$x_1(E) = y_0(D) = 0.2$$

HITS

- Compute hub value

$$
\begin{aligned}
y_1(A) &= 0 \\
y_1(B) &= x_1(A) + x_1(C) = 0.6 + 0.4 = 1 \\
y_1(C) &= x_1(A) = 0.6 \\
y_1(D) &= x_1(A) + x_1(B) + x_1(C) + x_1(E) = 0.6 + 0.2 + 0.4 + 0.2 = 1.4 \\
y_1(E) &= x_1(D) = 0.2
\end{aligned}
$$

HITS

- Normalize

$$x_1(A) = 0.75$$
$$x_1(B) = 0.25$$
$$x_1(C) = 0.50$$
$$x_1(D) = 0.25$$
$$x_1(E) = 0.25$$

$$y_1(A) = 0$$
$$y_1(B) = 0.546$$
$$y_1(C) = 0.327$$
$$y_1(D) = 0.764$$
$$y_1(E) = 0.109$$

HITS

- Second iteration

$$x_2(A) = y_1(B) + y_1(C) + y_1(D) = 0.546 + 0.327 + 0.764 = 1.637$$
$$x_2(B) = y_1(D) = 0.764$$
$$x_2(C) = y_1(B) + y_1(D) = 0.546 + 0.764 = 1.310$$
$$x_2(D) = y_1(E) = 0.109$$
$$x_2(E) = y_1(D) = 0.764$$

HITS

- Second iteration

$$y_2(A) = 0$$
$$y_2(B) = x_2(A) + x_2(C) = 1.637 + 1.310 = 2.947$$
$$y_2(C) = x_2(A) = 1.637$$
$$y_2(D) = x_2(A) + x_2(B) + x_2(C) + x_2(E) = 1.637 + 0.764 + 1.31 + 0.764 = 4.475$$
$$y_2(E) = x_2(D) = 0.109$$

HITS

- Normalize after second iteration

$$x_2(A) = 0.694$$
$$x_2(B) = 0.324$$
$$x_2(C) = 0.555$$
$$x_2(D) = 0.046$$
$$x_2(E) = 0.324$$

$$y_2(A) = 0$$
$$y_2(B) = 0.526$$
$$y_2(C) = 0.292$$
$$y_2(D) = 0.799$$
$$y_2(E) = 0.019$$

Q3-Memory-based CF

- Challenging:
  - Cold start problem
  - Data sparsity

One typical problem caused by the data sparsity is the cold start problem. As collaborative filtering methods recommend items based on users' past preferences, new users will need to rate sufficient number of items to enable the system to capture their preferences accurately and thus provides reliable recommendations.

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	0	2	5	3	1	0
U_2	3	5	4	3	0	2
U_3	4	0	1	4	2	2
U_4	3	0	4	5	5	3
U_5	1	3	5	0	2	2
U_6	3	0	0	0	0	0

Q3: Memory-based CF

- Pearson correlation coefficient

The Pearson correlation similarity of two users x, y is defined as

$$\mathrm{simil}(x,y) = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})(r_{y,i} - \bar{r_y})}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})^2 \sum_{i \in I_{xy}} (r_{y,i} - \bar{r_y})^2}}$$

where I_{xy} is the set of items rated by both user x and user y.

https://en.wikipedia.org/wiki/Collaborative_filtering

Q3: Memory-based CF

- Cosine similarity

$$\text{simil}(x,y) = \cos(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{||\vec{x}|| \times ||\vec{y}||} = \frac{\sum_{i\in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i\in I_x} r_{x,i}^2}\sqrt{\sum_{i\in I_y} r_{y,i}^2}}$$

Q3: Memory-based CF

- Top-k memory-based CF

$$r_{u,i} = \bar{r_u} + k \sum_{u' \in U} \text{simil}(u, u')(r_{u',i} - \bar{r_{u'}})$$

where k is a normalizing factor defined as $k = 1/\sum_{u' \in U} |\text{simil}(u, u')|$.

and $\bar{r_u}$ is the average rating of user u for all the items rated by u.

Q3: Memory-based CF

- PCC for U3:
  - 1: -0.327, 2: -0.272, 3:1, 4: 0, 5:-0.841, 6:0
- Top two similar users:
  - 4 and 6
- Prediction:
  - avg(u3) = 2.6

Q3: Memory-based CF

- Cos_similarity for U3 (intersection):
  - 1: 0.701, 2: 0.853, 3:1, 4: 0.886, 5:0.583, 6:1
- Top two similar users:
  - 4 and 6
- Prediction:
  - avg(u3) = 2.6

Q3: Memory-based CF

- Cos_similarity for U3 (count zero):
  - 1: 0.475, 2: 0.630, 3:1, 4: 0.886, 5:0.405, 6:0.625
- Top two similar users:
  - 2 and 4
- Prediction:
  - p= 2.6+1.6 = 4.2

Q3: Memory-based CF

- PCC for I5:
  - 1: 0.189, 2: 1, 3:-0.051, 4: 0.961, 5:1, 6:1
- Top two similar users:
  - 2 and 6
- Prediction (mean-center):
  - avg(I2) = 3.33, avg(I5) = 2.5, avg(I6) = 2.25
  - p = avg(I5) + [1*(5-avg(I2))+1*(2-avg(I6))]/2=3.21
- Prediction (direct)
  - p = (5+2)/2 = 3.5

Q3: Memory-based CF

- Cos_similarity for I5 (intersection):
  - I: 0.853, 2: 0.992, 3:0.775, 4: 0.929, 5:1, 6:0.971
- Top two similar users:
  - 2 and 6
- Prediction(mean-center):
  - avg(I2) = 3.33, avg(I5) = 2.5, avg(I6) = 2.25
  - p = avg(I5) + [0.992*(5-avg(I2))+0.971*(2-avg(I6))] /(0.992+0.971) = 3.22
- Prediction(direct):
  - p=(0.992*5+0.971*2)/(0.992+0.971) =3.516

Q3: Memory-based CF

- Cos_similarity for I5 (count zero):
  - I: 0.646, 2: 0.223, 3:0.697, 4: 0.804, 5:1, 6:0.861
- Top two similar users:
  - 4 and 6
- Prediction(mean-center):
  - avg(I4) = 3.75, avg(I5) = 2.5, avg(I6) = 2.25
  - p = avg(I5) + [0.804*(3-avg(I4))+0.861*(2-avg(I6))] /(0.804+0.861) = 2.009
- Prediction(direct):
  - p=(0.804*3+0.861*2)/(0.804+0.861) =2.483

Q4

- Square error

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2$$

Q4

- Answer:
  - Option 1 is better.
  - (1) sqe = 0.5, (2) sqe = 2.4

Midterm Overview

- Graph basics (hw2, 20pts)
  - Radius, diameter, center, cluster coefficient,
  - degree, density, degree sequence, degree distribution sequence
  - Adjacency matrix, path matrix, Laplacian matrix
  - Betweenness, closeness
- Ford-Fulkerson algorithm (hw3, 20pts)
  - How to implement
  - Advantage & disadvantage
- PageRank and HITS (hw3, 20pts)
  - How to implement
  - Understand the feature of the algorithms
- Memory-based recommendation (hw3, 20pts)
  - How to implement
  - How to use social information to improve recommender system
- Others (materials in your lectures, 20pts, choice question)

Grade Assessment Scheme

- Assignment: 20%
- Midterm: 30%
- Project: 50%