first name **last name** **id#**

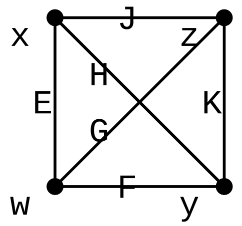
30 min **30 marks** **closed book** **no devices** **2 pages** **page 1**

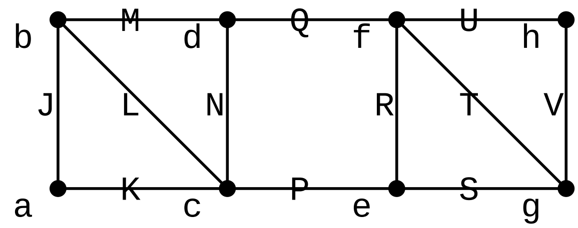
1. [10 marks]

   In the box at right,
   for `H = [[0,1,2],[1,0,2],[1,0,2]]`
   and `R = [[1,2,0],[0,1,2],[2,1,0]]`,
   show the output printed by `m=propose_reject(H,R)`.

```
def propose_reject(H,R):
  n = pref_system_size(H,R)
  F,C = [None] * n, [0 for j in range(n)]
  rejection = True
  while rejection:
    rejection = False
    for j in range(n):
      h_choice = H[j][C[j]] # current H proposal
      if F[h_choice] == None: #R has no prop'ls
        F[h_choice] = j
        print(' ',j,' prop ',h_choice,': maybe')
      elif F[h_choice] != j: #R has 2 prop'ls
        r_maybe = F[h_choice] #R's current prop'l
        if prefers(R[h_choice], j, r_maybe):
          r_reject, r_maybe = r_maybe, j
          F[h_choice] = r_maybe
        else:
          r_reject = j
        print(' ',j,'prop',h_choice,
              ':pref',r_maybe,',rej',r_reject)
        C[r_reject] += 1 # H[j_rej.]: next pref
        rejection = True # a prop'l was rejected
  P = [H[j][C[j]] for j in range(n)]
  print('\nj  P  C  F')
  [print(j, P[j], C[j], F[j]) for j in range(n)]
  return P
```

Show your rough work here.





Recall: a *cut* of a graph is a partition of the node set into two non-empty subsets. On the small graph (above left), {{w,x}, {y,z}} is a cut with cross-edges {F,G,H,J}. RKMC is the randomized Kruskal min cut algorithm: unless otherwise stated, its input is a uniform-random permutation of the edges.

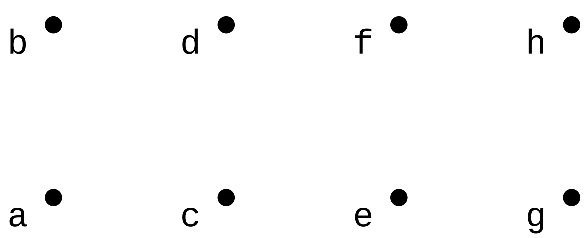
2. [6 marks] For the big graph, give two min cuts. (For part marks, give one min cut.)

   `cut node-set partition ______ ______` `cut cross-edges ________________`

   `cut node-set partition ______ ______` `cut cross-edges ________________`

3. [6 marks] On the nodes below, draw the final forest found by RKMC when edges are input in order QPRKNTUVSMLJ.

   Also, give the corresponding cut partition and cross-edges.



`cut node-set partition ______ _______`

`cut cross-edges ______________`

4. [2+6 marks]

   i) Let p be the probability that one execution of RKMC returns a min cut. By the theorem in class, for the graph above, we know that p is (circle one of these three) $\geq$ $=$ $\leq$ (now fill in) ________.

   ii) Let b be your bound from i). Which is true (circle one): $b < p$ $b = p$ $b > p$?

   Prove your answer.

first name **last name** **id#**

30 min **30 marks** **closed book** **no devices** **2 pages** **page 1**

1. [10 marks]

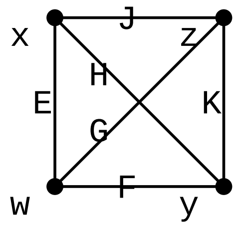
   In the box at right,
   for `H = [[0,1,2],[1,0,2],[1,0,2]]`
   and `R = [[1,2,0],[0,1,2],[2,1,0]]`,
   show the output printed by `m=propose_reject(H,R)`.

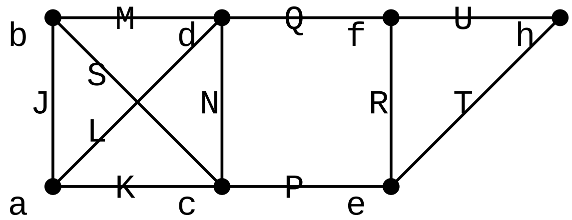
```
def propose_reject(H,R):
  n = pref_system_size(H,R)
  F,C = [None] * n, [0 for j in range(n)]
  rejection = True
  while rejection:
    rejection = False
    for j in range(n):
      h_choice = H[j][C[j]] # current H proposal
      if F[h_choice] == None: #R has no prop'ls
        F[h_choice] = j
        print(' ',j,' prop ',h_choice,': maybe')
      elif F[h_choice] != j: #R has 2 prop'ls
        r_maybe = F[h_choice] #R's current prop'l
        if prefers(R[h_choice], j, r_maybe):
          r_reject, r_maybe = r_maybe, j
          F[h_choice] = r_maybe
        else:
          r_reject = j
        print(' ',j,'prop',h_choice,
              ':pref',r_maybe,',rej',r_reject)
        C[r_reject] += 1 # H[j_rej.]: next pref
        rejection = True # a prop'l was rejected
  P = [H[j][C[j]] for j in range(n)]
  print('\nj  P  C  F')
  [print(j, P[j], C[j], F[j]) for j in range(n)]
  return P
```

Show your rough work here.

first name **last name** **id#**

30 min **30 marks** **closed book** **no devices** **2 pages** **page 2**





Recall: a *cut* of a graph is a partition of the node set into two non-empty subsets. On the small graph (above left), {{w,x}, {y,z}} is a cut with cross-edges {F,G,H,J}. RKMC is the randomized Kruskal min cut algorithm: unless otherwise stated, its input is a uniform-random permutation of the edges.

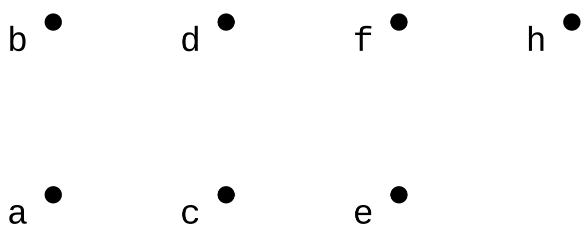
2. [6 marks] For the big graph, give two min cuts. (For part marks, give one min cut.)

`cut node-set partition ______ ______` `cut cross-edges ________________`

`cut node-set partition ______ ______` `cut cross-edges ________________`

3. [6 marks] On the nodes below, draw the final forest found by RKMC when edges are input in order QPMUJSLTNKR.

   Also, give the corresponding cut partition and cross-edges.



`cut node-set partition ______ _______`

`cut cross-edges ______________`

4. [2+6 marks]

   i) Let p be the probability that one execution of RKMC returns a min cut. By the theorem in class, for the graph above, we know that p is (circle one of these three) $=$ $\leq$ $\geq$ (now fill in) ________.

   b) Let c be your bound from i). Which is true (circle one): $c > p$ $c = p$ $c < p$?

   Prove your answer.

first name **last name** **id#**

30 min **30 marks** **closed book** **no devices** **2 pages** **page 1**

1. [10 marks]

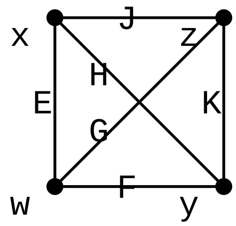
   In the box at right,
   for `H = [[0,1,2],[1,0,2],[1,0,2]]`
   and `R = [[1,2,0],[0,1,2],[2,1,0]]`,
   show the output printed by `m=propose_reject(H,R)`.

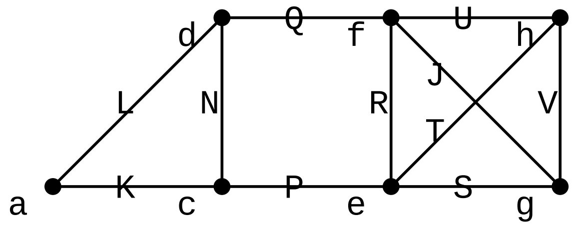
```
def propose_reject(H,R):
  n = pref_system_size(H,R)
  F,C = [None] * n, [0 for j in range(n)]
  rejection = True
  while rejection:
    rejection = False
    for j in range(n):
      h_choice = H[j][C[j]] # current H proposal
      if F[h_choice] == None: #R has no prop'ls
        F[h_choice] = j
        print(' ',j,' prop ',h_choice,': maybe')
      elif F[h_choice] != j: #R has 2 prop'ls
        r_maybe = F[h_choice] #R's current prop'l
        if prefers(R[h_choice], j, r_maybe):
          r_reject, r_maybe = r_maybe, j
          F[h_choice] = r_maybe
        else:
          r_reject = j
        print(' ',j,'prop',h_choice,
             ':pref',r_maybe,',rej',r_reject)
        C[r_reject] += 1 # H[j_rej.]: next pref
        rejection = True # a prop'l was rejected
  P = [H[j][C[j]] for j in range(n)]
  print('\nj  P  C  F')
  [print(j, P[j], C[j], F[j]) for j in range(n)]
  return P
```

Show your rough work here.

first name **last name** **id#**

30 min **30 marks** **closed book** **no devices** **2 pages**





Recall: a *cut* of a graph is a partition of the node set into two non-empty subsets. On the small graph (above left), {{w,x}, {y,z}} is a cut with cross-edges {F,G,H,J}. RKMC is the randomized Kruskal min cut algorithm: unless otherwise stated, its input is a uniform-random permutation of the edges.

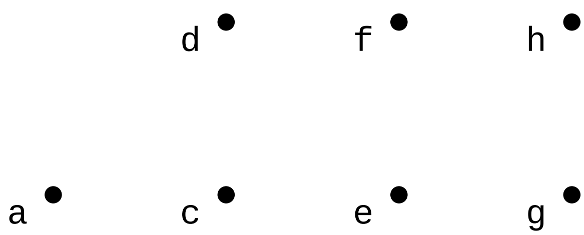
2. [6 marks] For the big graph, give two min cuts. (For part marks, give one min cut.)

`cut node-set partition ______ ______` `cut cross-edges ________________`

`cut node-set partition ______ ______` `cut cross-edges ________________`

3. [6 marks] On the nodes below, draw the final forest found by RKMC when edges are input in order QPRNVLUJKTS.

   Also, give the corresponding cut partition and cross-edges.



`cut node-set partition ______ _______`

`cut cross-edges ______________`

4. [2+6 marks]

   i) Let p be the probability that one execution of RKMC returns a min cut. By the theorem in class, for the graph above, we know that p is (circle one of these three) $\leq$ $\geq$ $=$ (now fill in) ________.

   b) Let d be your bound from i). Which is true (circle one): $d = p$ $d < p$ $d > p$?

   Prove your answer.