ualberta cmput 204, fall 2014, section a2/ea2 (hayward)
final exam 3 hours december 16

---

last name________________________ (no student id# this page)

first name________________________

- print your name above
- **do not detach** any page from the staple   this exam has 7 physical pages (cover plus 6 exam pages)
- 6 exam pages, 8 marks/page, 48 marks total
- **write legibly**: use available space (and back of pages if needed) for your solutions
- **to receive part marks for an incorrect answer, you must show your work**
- closed book: no notes, no electronic devices: pen/pencil only

for instructor use only: do not write below

total marks	question	your marks
8	1	
8	2	
8	3	
8	4	
8	5	
8	6	
48		

last name ________________________ **id#**________________________

1. (i) Show the output from `foo(15,27)`.

```
def foo(x,y): # y >= 1
  if y==1:
    print 'foo',x,y,'=',x
    return x
  elif 0== y%2:
    z = 2*foo(x,y/2)
    print 'foo',x,y,'=', z
    return z
  else:
    z = x+2*foo(x,y/2)
    print 'foo',x,y,'=', z
    return z
```

(ii) Let $f(n)$ be the number of lines of output from `foo(15,n)`. Complete the table.

n	1	2	3	4	5	6	7	8	9	10
$f(n)$										

Give a recurrence relation: $f(n) =$ ________________________________

____

Solve the recurrence relation. Give the simplest answer. $f(n) = \Theta($ ____________________ $)$

(iii) Let t be an integer $t \geq 1$. For all integers x, for all integers z with $1 \leq z \leq t$, assume that `foo(x,z)` returns x times z. Prove/disprove: `foo(x,t+1)` returns x times $t+1$.

last name ________________________ **id#**________________________

2.(i) Let x and y be positive integers. Assume that each is small enough to fit into one memory location. How long does it take to determine whether $x < y$?

time = $\Theta($ ____________________________ $)$

justification: ____

(ii) P is a list of n integers, each small enough to fit into one memory location. Give the runtime of this code.

```
min = P[0]
for j in range(1, len(P)):
  if P[j] < min: min = P[j]
```

runtime = $\Theta($ ____________________________ $)$

justification: ____

(iii) Repeat (i), assuming x and y are arbitrarily large. (Hint: how many bits are needed to represent x? y?)

time = $\Theta($ ____________________________ $)$

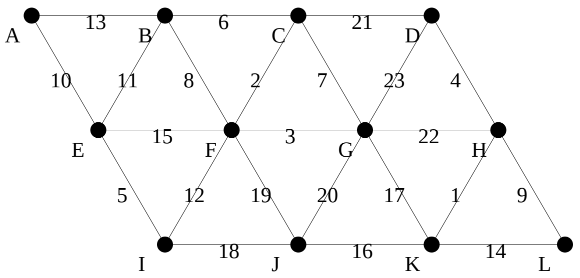
justification: ____

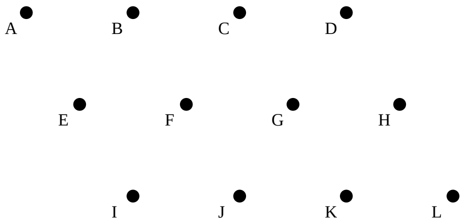
(iv) Let z be arbitrarily large. Repeat (ii), assuming each number in P is in $\{z, z+1, \ldots, z+1000\}$.

runtime = $\Theta($ ____________________________ $)$

justification: ____

last name ________________________ id#________________________





3. (i) G is the above graph. Show the output from `prim(G,'A')`, and draw the MST on the above nodes.

output:

```
def indexOfMin(L,C):
  ndx = 0
  for j in range(1,len(L)):
    if C[L[j]] < C[L[ndx]]: ndx = j
  return ndx

def prim(G,source): # Prim's MST
  cost, parent, fringe = {}, {}, []
  for v in G:
    cost[v], parent[v] = infinity(), -1
    fringe.append(v)
  cost[source],parent[source] = 0, source

  while len(fringe)>0:
    u = fringe.pop(indexOfMin(fringe, cost))
    if u != source: print 'edge',u,parent[u]
    for (v,costuv) in G[u]:
      if cost[v] > costuv:
        cost[v] = costuv
        parent[v] = u

G = {'A': [['B',13],['E',10]],
     'B': [['A',13],['C',6],['E',11],['F',8]],
      ...
```

Now assume that the input graph G has n nodes and m edges, and that each number used in the algorithm is small enough to fit into one memory location, and that the graph is represented with adjacency lists.

(ii) As a function of n and m give the runtime for `prim(G,'A')`: $O($ ______________________________ $)$

justification: ____

____

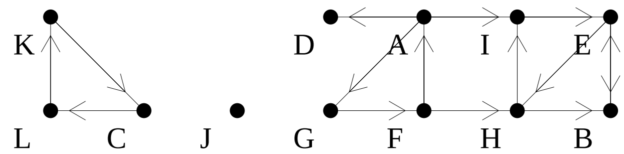
(iii) Repeat (ii) if `cost` is stored in a binary-heap priority-queue: $O($ ______________________________ $)$

justification: ____

____

last name ________________________ **id#**________________________

4.



(i) G is the above digraph. T is the transpose of G. Give the adjacency list representation for T. List the neighbours of each vertex in alphabetic order.

`A:`	`E:`	`I:`
`B:`	`F:`	`J:`
`C:`	`G:`	`K:`
`D:`	`H:`	`L:`

(ii) Starting from A, and (in the outermost loop) processing vertices in alphabetic order, draw the recursion forest of a depth-first search traversal of T. Draw back edges and cross edges using dashed lines.

(iii) To find the strongly connected components of G, first

order the vertices in ____.

Then, using this order, ____.

(iv) List the strongly connected components of G.

last name ________________________ id#________________________

5. (i) For each element of S, find the length of a longest **decreasing** subsequence ending with that element.

S	7	10	6	2	4	8	9	3	2	7	9	1
length												

(ii) Matrices A, B, C, D have respective dimensions 5×3, 3×7, 7×1, 1×6. Show how to compute the matrix product $A \times B \times C \times D$, so as to minimize the total number of scalar multiplications: give a best parenthesization, and the total number of scalar multiplications.

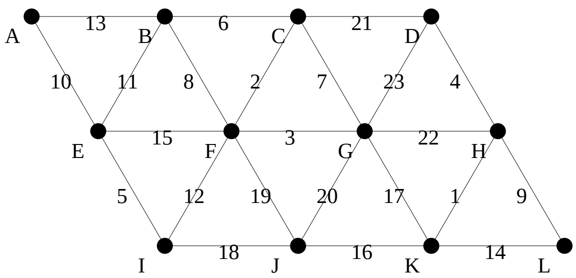
```
        A        B        C        D
     - - - - - - - - - - - -
    |        |        |        |        |
  A |        |        |        |        |
    |        |        |        |        |
     - - - - - - - - - - - -
             |        |        |        |
  B          |        |        |        |
             |        |        |        |
              - - - - - - - - -
                      |        |        |
  C                   |        |        |
                      |        |        |
                       - - - - - -
                               |        |
  D                            |        |
                               |        |
                                - - -
```

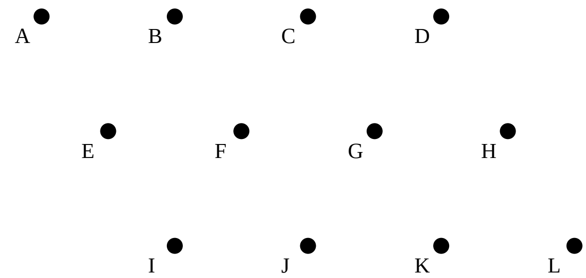
best parenthesization ______________________________

minimum total number of scalar multiplications ______________________________

last name ________________________ **id#**________________________

6. (i) Recall: a cut of a graph is a partition of the vertices into two parts; the size of the cut is the **number of edges** with one end in each part. Karger's min cut algorithm runs on the weighted graph below, **but** the random edge selection procedure is broken: instead of selecting a random edge, it selects the minimum weight edge not yet selected (so the first edge selected would be HK).





On the nodes, draw the Karger forest. Also, give the cut found: ________________________________

(ii) Give a min cut of this graph, and explain briefly how you know it is minimum.

a min cut is ________________________________

justification: ____

Now for (iii) and (iv), assume that the random edge selection procedure in Karger's algorithm works properly. For the graph above, let p be the probability that the cut found in one run of Karger's algorithm is a min cut.

(iii) Give a bound on p.

$p \geq$ ________________________

justification: ____

(iv) You run Karger's algorithm 3 times on the graph above. As an expression in p, what is the probability that at least one of the 3 cuts found is a min cut?

prob that at least one of the 3 cuts is min $\geq$ ________________________

justification: ____