cmput 204, fall 2013, section A2/EA2 (hayward)
final examination (3 hours)
December 16, 2013

computing science, UAlberta

Last Name:________________________ **(No student ID this page!)**

First Name:________________________

- put name above print clearly
- **do not detach** any page from the staple this exam has 7 physical 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 on this page

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

Last Name:________________________ **ID#:**________________________

1.

```
A -17-- B --4-- C --1-- D
|       |       |       |
6       8       1       1
|       |       |       |
E --1-- F --5-- G --1-- H
|       |       |       |
10      9       6       8
|       |       |       |
I --2-- J --2-- K --7-- L
|       |       |       |
2       2       2       3
|       |       |       |
M --2-- N --2-- O --1-- P
```

a) Kruskal's MST algorithm runs on the above weighted graph. Each set of equal-weight edges is considered in alphabetic order. Above: list the edges of the output MST, in the order they are found.

b) Kruskal's MST algorithm runs on the above weighted graph, but this time each set of equal-weight edges is considered in random order. Give the number of different MSTs that can be returned, if this algorithm is run an unlimited number of times. Justify your answer.

Last Name:________________________ **ID#:**________________________

2.

```
A -17-- B --4-- C --1-- D                 A       B       C       D
|       |       |       |
6       8       1       1
|       |       |       |
E --1-- F --5-- G --1-- H                 E       F       G       H
|       |       |       |
10      9       6       8
|       |       |       |
I --2-- J --2-- K --7-- L                 I       J       K       L
|       |       |       |
2       2       2       3
|       |       |       |
M --2-- N --2-- O --1-- P                 M       N       O       P

vertex           A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P

distance from A: _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _

order in which edges found:
```

a) Prims's single-source shortest path algorithm runs on the above weighted graph, with source A. Each node's neighbourhood list is stored in alphabetic order. Above, show the output: for each vertex, give the distance from A. Also: on the right-hand set of nodes, draw the edges of the shortest-path tree, and label these edges in the order in which they are added to the tree.

b) In a weighted graph G, let e be an edge of minimum weight. Prove or disprove: some MST contains e.

Last Name:________________________ **ID#:**________________________

3.

```
def test(G):
  s = {}
  for v in G: s[v] = False
  for v in G:
    if not s[v]:
      ex(G,v,s)
      print ''

def ex(G,v,s):
  s[v] = True
  print v,
  for w in G[v]:
    if not s[w]:
      ex(G,w,s)
  print v,

G = {'A':['I','E'], 'B':[], 'C':['F','E','D'], 'D':['H'], 'E':['G'],
     'F':['D','C'], 'G':['C','B'], 'H':['F','B'], 'I':['D','A']}
test(G)
```

a) Show the output of the above program.

b) Using Θ notation, for an input graph G with n nodes and m edges, give the runtime of this program as a function of n and m. Assume that n and m are small enough so that basic operations can be executed in constant time. Justify your answer.

4. One way to build a heap (here, a min-heap, so smallest element at root) of size n is to start with a compact binary tree, and then, for j from $\lfloor n/2 \rfloor$ down to 1, for the subtree rooted at position j, turn the almost-heap rooted at position j into a heap.

```
      9
    /   \
  3       7
 / \     / \
8   4   1   5
```

a) Show the heap created from the above compact binary tree: above, draw the final heap beside the tree.

b) For $n = 31$, give the worst-case number of key comparisons to build a heap this way. Justify briefly.

Last Name:________________________ **ID#:**________________________

5. For each element of sequence S, find the length of a longest increasing subsequence ending with that element.

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

b) Matrices A, B, C, D have respective dimensions 5×7, 7×3, 3×6, 6×2. 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.

Last Name:________________________ **ID#:**________________________

---

6. The following code is related to worst-case performance of a recursive solution to the longest increasing subsequence.

```
def Y(n):
  print 'call Y(',n,')'
  if n>0:
    for j in range(n): #so j runs from 0 to n-1
      Y(j)
```

a) Below, in each column, are the outputs from the calls `Y(0)` and `Y(1)`. Show the output from the calls `Y(2)` and `Y(3)`.

`call Y(0)`	`call Y(1)`
	`call Y(0)`

b) Let $f(n)$ be the number of lines of output from the call $Y(n)$. Find a simple expression for $f(n)$, and prove by induction on n that your expression is correct.

End of exam.