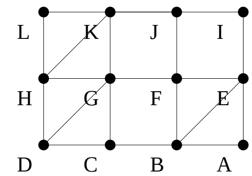
last name **first names** **id#**

cmput 204 **assignment 3** **due start of class, 2015 oct 26**

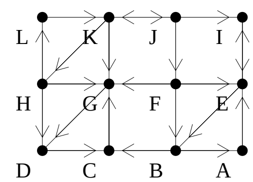
Write your answers on a copy of this document. Hand in 4 separate pages, not stapled.

1. (a) **If you leave this question blank, your assignment will not be marked and its weight will be transferred to the final exam.** Acknowledge **all** sources, including all references and all people with whom you discussed any part of any question (for each discussion, list the relevant questions):



(b) Trace bfs on this graph:
List nodes in bfs-order (the order seen by bfs).

Draw the bfs traversal forest.



Trace bfs on this digraph:
List nodes in bfs-order.

Draw the bfs traversal forest.

(c) A graph has (dfs) preorder ACGILMHJKNOBDEIF and postorder LMIGJNOKHCADIEFB. Draw the dfs traversal forest. Hint: for each subtree traversal, root is first/last in pre/postorder.

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

2. (a) Let $K(n)$ be the number of key comparisons performed by a mergesort of n keys. Explain briefly why $K(n)$ satisfies the following recurrence relation:

$$\begin{aligned} K(n) &= n - 1 + K(\lfloor n/2 \rfloor) + K(\lceil n/2 \rceil) \text{ if } n \geq 2 \\ &= 0 \text{ if } n \leq 1 \end{aligned}$$

(b) Let $B(n)$ be the lower bound on the number of key comparisons required by a comparison-based sort of n keys. Explain briefly why $B(n) = \lceil lg(n!) \rceil$.

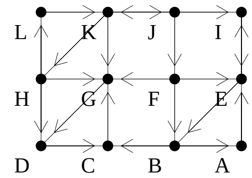
(c) Give an $O(n \log n)$ time python program to compute $K(n)$ exactly. Explain briefly why your program has this runtime.

(d) Compute $K(n)$ and $B(n)$ exactly for $n = 1, 10, 100, \ldots, 10^7$.

(e) For large n, assume $K(n) \approx kn \lg n$ and $B(n) \approx bn \lg n$, where k and b are constants. Using your data, approximate k and b.

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

3. (a)



Trace the scc algorithm on this digraph. Give postorder of transpose:

For the last node x in the above postorder, give the scc containing x:

Using the reverse of the above postorder, draw the dfs traversal forest:

(b) Let D be a digraph with transpose T. Let S be a strongly connected component of D. Let x, y be nodes of S. Is there a path from x to y in D? Explain briefly.

Is there a path from y to x in D? Explain briefly.

Is there a path from x to y in T? Explain briefly.

Is there a path from y to x in T? Explain briefly.

Is S a strongly connected component of T? Explain briefly.

4. (a)
```
def qsort(list, start, end):
  print start,end,
  if start < end:
    print list
    split = pn(list, start, end)
    qsort(list, start, split-1)
    qsort(list, split+1, end)
  else: print ''

L = [44,88,11,0,33,99,22,77,66,55]
qsort(L,0,len(L)-1)
```

Assume (as in seminar 4) that `partition( )` uses the first list element as pivot, and preserves the relative order of the two sublists. For the above code:

Draw the `qsort( )` recursion tree, labelling each node with start and end. So, the root node is (0,9).

Show the output.

(b) Usually, `partition( )` does *not* preserve the relative order of the two sublists. Briefly, explain why.

(c) A version of quicksort performs, on average, $C(n)$ key comparisons to sort n keys, where $C(n) = 0$ for $n \leq 1$, and for $n \geq 2$, $C(n) = 2(n+1)\sum_{j=1}^{n} 1/j - (17n+5)/6$. Compute $C(n)$ for $n = 1, 10, 100, \ldots, 10^7$, and compare with $K(n)$ and $B(n)$ from question 2.