Tentamen i kursen **Datastrukturer och algoritmer**
Totalt antal uppgifter: 7 st
Lärare: Jingsen Chen (0920-49 20 44)

Kurskod	D0041D
Tentamensdatum	2019-03-29
Skrivtid	4 timmar

Tillåtna hjälpmedel: språk ordbok

1. Given numbers $12, 19, 40, 22, 20, 29$,

   (a) Sort these numbers using *mergesort* step by step. **(5p)**

   (b) Sort these numbers using *insertionsort* step by step. **(5p)**

   (c) Mergesort will always perform fewer comparisons than insertionsort when applied to the same input array. Is this statement TRUE or FALSE? Why? No credit will be given without justifications. **(5p)**

   (d) Show the result of insert all the numbers above *in that order* into an initially empty hash table of size 10 with the hash function $h(x, i) = (x \bmod 10 + i) \bmod 10$, $i = 0, 1, 2, \cdots, 9$. **(5p)**

2. The preorder traversal of a binary tree first visits the root and then recursively visits the left subtree followed by the right subtree of the root.

   (a) Given an array $\langle 14, 2, 1, 5, 4, 16, 15\rangle$ that represents the preorder traversal of a binary search tree, draw this tree. **(5p)**

   (b) Given an array $\langle 1, 2, 4, 5, 14, 15, 16\rangle$ that represents the preorder traversal of a min-heap, draw this heap. **(5p)**

3. For each of the following statements, indicate whether the statement is TRUE or FALSE. **Justify your answers**. *That is, if the statement is correct, state why; and if the statement is wrong, give a counter-example.* No credit will be given without justifications.

   (a) Binary search on a sorted array of length n takes $O(n)$ time in the worst case. **(3p)**

   (b) Inserting an element into a binary search tree of size n takes $\Omega(\log n)$ time in the worst case. **(3p)**

   (c) If an algorithm A runs in $\Omega(n \log n)$ time in the worst case, then the time needed by A on any input of size n is $\Theta(n \log n)$. **(3p)**

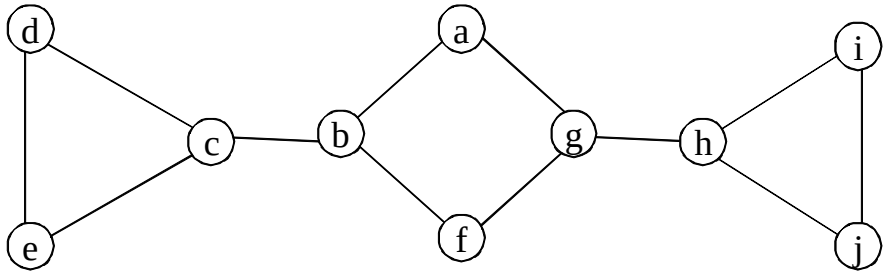
   (d) A spanning tree of a connected and undirected graph $G = (V, E)$ can be found in $O(\|E\|)$ time. **(3p)**

   (e) Let G be a connected directed graph with edge-weights. If some of the edge-weights are negative, then Dijkstra´s algorithm is asymptotically slower than Bellman-Ford algorithm. **(3p)**

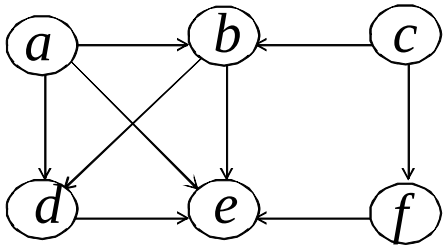
4. This question is about graph algorithms.

   (a) Draw the adjacency list and the adjacency matrix representation of the graph, respectively. **(5p)**

   (b) Draw a depth-first search tree and a breadth-first search tree of the graph below starting at the node a, respectively. **(5p)**



(c) Compute a topological ordering of the following graph. **(5p)**



5. Given an unsorted array $A = \langle a_1, a_2, \cdots, a_n \rangle$ of n numbers, which may be positive, negative, or zero. Design a *divide and conquer algorithm* to compute the largest suffix sum of A; that is, computing the value $\max_{1 \le j \le n}\{a_j + \cdots + a_n\}$. Your algorithm should run in $O(n)$ time in the worst case. **(10p)**

   For example, if $A = \langle 2, -3, 5, -4, 3 \rangle$, then all the suffix sums of A are:

   $a_1 + a_2 + a_3 + a_4 + a_5 = 2 + (-3) + 5 + (-4) + 3 = 3$
   $a_2 + a_3 + a_4 + a_5 = (-3) + 5 + (-4) + 3 = 1$
   $a_3 + a_4 + a_5 = 5 + (-4) + 3 = 4$
   $a_4 + a_5 = (-4) + 3 = -1$
   $a_5 = 3$

   Hence, the largest suffix sum of A is 4 ($= a_3 + a_4 + a_5 = 5 + (-4) + 3$).

6. Show how to use data structure heap to merge k sorted lists into one sorted list in $O(n \log k)$ time in the worst case, where n is the total number of elements in all the input lists. **(10p)**

7. Given an unsorted array $A = \langle a_1, a_2, \cdots, a_n \rangle$ of n distinct numbers,

   (a) Design an algorithm to count the number of pairs (a_i, a_j), $1 \le i < j \le n$, for which $a_i + 1 = a_j$. For example, for the array $\langle 2, 8, 4, 10, 3, 5 \rangle$, there are two such pairs $(2, 3)$ and $(4, 5)$. Your algorithm should runs in $O(n)$ time in the average case. Notice that the distribution of the input numbers is unknown. **(10p)**

   (b) Show how to compute the smallest, the 2^{nd} smallest, the 4^{th} smallest, the 8^{th} smallest, $\cdots$, the $(\frac{n}{4})^{th}$ smallest, and the $(\frac{n}{2})^{th}$ smallest elements in $O(n)$ time in the worst case. **(10p)**