Tentamensdatum: 2020-08-28, 09:00 - 13:00
Tillåtna hjälpmedel: Engelsk-svensk ordbok
Totalt antal uppgifter: 9 st, 100p
Jourhavande lärare: 0920 / 49 20 44

1. Sort the following numbers step by step:

   (a) Sort $24, 13, 17, 34, 15, 27, 15, 29, 22, 26$ using *mergesort*; **(5p)**

   (b) Sort $24, 13, 17, 34, 15$ using *heapsort*. **(5p)**

   (c) What is the worst-case running time of *mergesort* and *heapsort* on a sorted sequence of length n (using Θ-notation), respectively. **(5p)**

2. Find recurrences describing the number of comparisons used by the following algorithms in the worst case.

   (a) In order to *sort* an input sequence of length n, we first divide the input into three parts of equal length and then sort each part recursively. If the length n of the input sequence is small, say $n \leq 3$, just sort the input using some sorting method. Thereafter, these three sorted subsequences are merged into a sorted output of size n. **(4p)**

   (b) A heap of size n can be constructed by using the recursive method below. First, a heap on the left subtree of the root and a heap on the right subtree of the root are built recursively. Then, these two heaps built can be merged together with the root into a heap on n elements by using $2 \log n$ comparisons. **(4p)**

3. a) Draw the graph with the following adjacency matrix: **(2p)**

	a	b	c	d	e
a	$-$	2	1	$-$	$-$
b	2	$-$	-1	5	$-$
c	1	-1	$-$	4	3
d	$-$	5	4	$-$	-1
e	$-$	$-$	3	-1	$-$

   b) Executing Prim's and Kruskal's minimum spanning tree algorithms, respectively, on the above graph. Number the edges in order of adding to the minimum spanning trees by the algorithms. **(4p)**

   c) In order to solve the single-source shortest path problem, which one(s) of Dijkstra's algorithm and Bellman-Ford's algorithm can be applied to the above graph. Justify your answers. No credit will be given without justifications. **(4p)**

4. 1) Draw the binary search tree that results when the keys $24, 13, 17, 34, 15$ are inserted in that order into an initially empty tree. **(4p)**

   2) Using the hash functions $h(x) = x \bmod 10$, show the result of inserting the keys $24, 13, 14, 34, 15$ into an initially empty table of size 10 using hashing with separate chaining. **(4p)**

3) For each of the following statements, indicate whether it is TRUE or FALSE, and justify your answers. That is, if the statement is TRUE, state why; and if the statement is FALSE, give a correct (corresponding) statement.

(a) An insertion in a binary search tree of size n can be done in $O(\log n)$ time in the worst case. **(3p)**

(b) The size of a hash table using double hashing must be a prime number. **(3p)**

(c) The minimum element in a binary search tree is always the leaf of the search tree. **(3p)**

5. Let **S** be an array of n elements. Each element in **S** is colored with either red or blue. The task is to rearrange the array so that all the red elements precede all the blue ones. Your algorithm should be in-place and run in $O(n)$ time in the worst case. **(10p)**

6. Design an $O(n)$-time algorithm that determines whether or not a given undirected graph on n vertices contains a cycle. Assume that the graph is given by its adjacency list. **(10p)**

7. Given a set S of n distinct integers (all the integers are between 0 and $n^2 - 1$). Design a worst-case linear-time algorithm that can answer queries whether S contains two elements x and y such that $x + y = c$ for any given number c. **(10p)**

8. An *anagram* of a word W is another word made up of the same letters as W. For example, stop, tops, and post are anagrams of each other. Given a set of words, design an efficient algorithm to make a list for each word of all its anagrams that appear in the set. Let n denote the sum of lengths of the words in the set. If we count only the number of letter-letter comparisons used, your algorithm should runs in $O(n)$ time and space in the worst case. **(10p)**

9. *Arbitrage* is the use of discrepancies in currency exchange rates to transform one unit of a currency into more than one unit of the same currency. For example, suppose that 1 US dollar buys 10,52 Swedish crown, 1 Swedish crown buys 11,36 Japanese yen, and 1 Japanese yen buys 0,0084 US dollars. Then, by converting currencies, a trader can start with 1 US dollar and buy $10,52 \times 11,36 \times 0,0084 = 1,00386$ US dollars, thus turning a profit of 0,386 percent.

   Suppose that we are given n currencies $C_1, C_2, ..., C_n$ and an $n \times n$ table R of exchange rates, such that one unit of currency C_i buys $R[i, j]$ units of currency C_j. Design an efficient to determine whether or not there exists a sequence of currencies $\left\langle C_{i_1}, C_{i_2}, ..., C_{i_k} \right\rangle$ such that $R[i_1, i_2] \times R[i_2, i_3] \times ... \times R[i_{k-1}, i_k] \times R[i_k, i_1] > 1$. Your algorithm should run in $O\left(n^3\right)$ time in the worst case. **(10p)**