Re-Examination

Algorithms, Data Structures and Software Engineering for Media Technology

Medialogy 8th semester

18 August 2022, 09.00 – 13.00

Name:_____

Cpr.no.: _____

Study no.: _____

Continued on next page

Aalborg University | MSc in Medialogy | Med8

Algorithms, Data Structures and Software Engineering for Media Technology

Re-exam

August 2022

Instructions

- You have 4 hours to complete this examination.
- Neither electronic devices nor written material are allowed in the examination room.
- This examination consists of 10 questions. Each question is worth 10 marks. You must obtain at least 50 marks to pass.
- Do not write any answers on this question paper—answers written on the question paper will be ignored by the examiner. Write all your answers on the writing paper provided.
- Do not write your answers in pencil and do not use a pen with red or green ink. Use a pen with blue or black ink.
- Hand in no more than one answer to each question.
- Do not turn over until you are told to do so by the invigilator.

Continued on next page

Question 1

For each of the following equations, state whether it is true or false.

a) $n^2 = O(n^2)$

b) $n^3 \log(\log n) = \Theta(n^3 \log n)$

c) $n^2 = \Omega(n^2 \log n)$

d) $\sqrt{n} = \omega(n^{0.5})$

e) $\frac{1}{\sqrt{n}} = o(\log n)$

f) $n \log(n^5) = O\left(\left(\sqrt{n}\right)^3\right)$

g) $\sqrt{n} = \Theta(\log n)$

h) $n^{2.5} = \Omega\left(\left(\sqrt{n}\right)^5\right)$

i) $n \log n = \omega(n)$

j) $n^2 - n = o(n^2)$

[1 mark for each correct part]

Question 2

The Master Theorem is stated as follows:

Theorem 4.1 (Master theorem)
Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence

$$T(n) = aT(n/b) + f(n) ,$$

where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ has the following asymptotic bounds:

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$. ■

Given the Master Theorem, as stated above, write down the order of growth in terms of Θ notation for each of the following recurrences.

a) $T(n) = \sqrt{2}T\left(\frac{n}{2}\right) + \sqrt{n}$

b) $T(n) = 27T\left(\frac{n}{3}\right) + n^2$

c) $T(n) = 2T\left(\frac{n}{4}\right) + n$

d) $T(n) = 9T\left(\frac{n}{3}\right) + n^2$

e) $T(n) = 5T\left(\frac{n}{\sqrt{5}}\right) + n$

[2 marks for each correct part]

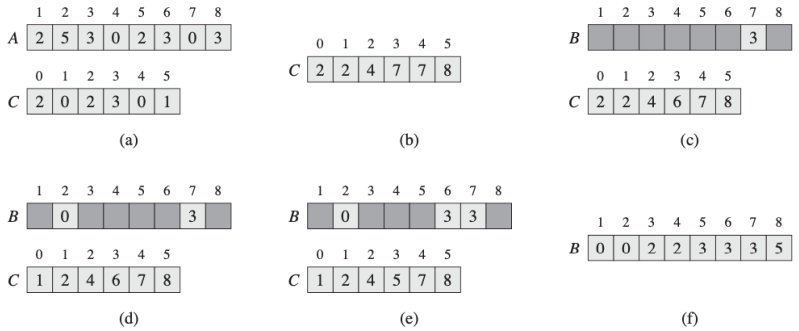
Continued on next page

Question 3

a) Using asymptotic notation, write down the optimal worst-case running time for a comparison sort algorithm.

b) Give two examples of comparison sort algorithms whose worst-case running times are optimal.

c) The following figure shows pseudocode for the counting sort algorithm along with an example of running the algorithm in parts (a) to (f) of the figure.




```
COUNTING-SORT(A, B, k)
 1  let C[0..k] be a new array
 2  for i = 0 to k
 3      C[i] = 0
 4  for j = 1 to A.length
 5      C[A[j]] = C[A[j]] + 1
 6  // C[i] now contains the number of elements equal to i.
 7  for i = 1 to k
 8      C[i] = C[i] + C[i − 1]
 9  // C[i] now contains the number of elements less than or equal to i.
10  for j = A.length downto 1
11      B[C[A[j]]] = A[j]
12      C[A[j]] = C[A[j]] − 1
```

i. Which of the three arrays used in the Counting Sort algorithm is used to hold the final sorted output?

ii. In part (b) of the figure above, what do the entries in the array *C* tell us about the input sequence?

d) Given an input array, *A*, in which each element is a *d*-digit number, write down pseudocode for the *radix sort* algorithm.

e) Why does the sorting algorithm *used by* radix sort have to be *stable*?

[2 marks for each correct part]

Continued on next page

Question 4

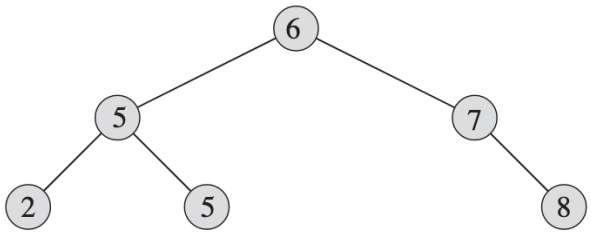
Suppose we are given a list of n items in random order and we store them in a hash table. Write down the following using asymptotic notation:

a) the worst-case running time for searching for an item in the hash table;
b) the worst-case running time for inserting a new item into the hash table;
c) the worst-case running time for deleting an item from the table, assuming we have already found it;
d) the worst-case run running time for deleting an item from the table, assuming we have *not* already found it;
e) the expected time for searching for an item in the hash table, assuming we have simple uniform hashing.

[2 marks for each correct part]

Question 5

a) What is the expected height of a randomly built binary search tree containing n items? Express your answer using asymptotic notation.
b) The following figure shows a binary search tree, with the key of each node printed in the node. Below the tree, pseudocode is provided for the inorder-tree-walk algorithm. This algorithm takes a node, x, as its argument. A node, x, has four properties: $x.key$ returns the key of x, $x.left$ returns the left child of x, $x.right$ returns the right child of x, and $x.parent$ returns the parent of x. What does the Inorder-Tree-Walk algorithm return when x refers to the root of the tree shown in the diagram?

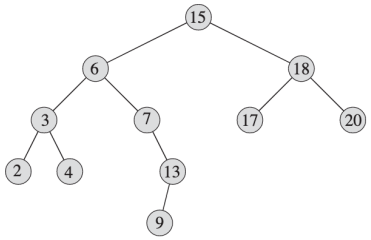



```
INORDER-TREE-WALK(x)
1  if x ≠ NIL
2      INORDER-TREE-WALK(x.left)
3      print x.key
4      INORDER-TREE-WALK(x.right)
```

c) What is the running time of the inorder tree walk algorithm shown in the figure above?

Continued on next page

d) The diagram below shows another binary search tree along with pseudocode for the iterative tree search algorithm. What is the worst-case running time of this algorithm assuming the input is of size n?




```
ITERATIVE-TREE-SEARCH(x, k)
1  while x ≠ NIL and k ≠ x.key
2      if k < x.key
3          x = x.left
4      else x = x.right
5  return x
```

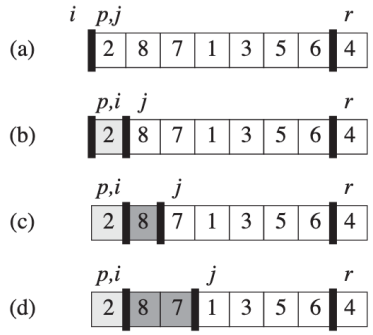
e) What is the expected running time of the iterative tree search algorithm shown above, assuming the tree is randomly built?

[2 marks for each correct part]

Question 6

The following diagram shows pseudocode for the Partition algorithm, as used in the Quicksort algorithm, along with the state of the variables in the algorithm after the first few iterations of the 'for' loop in lines 3 – 6 (diagrams (a) to (d)). Draw diagrams like the ones in (a) to (d), that illustrate the state of the variables in the algorithm after the remaining iterations of the 'for' loop.

```
PARTITION(A, p, r)
1  x = A[r]
2  i = p − 1
3  for j = p to r − 1
4      if A[j] ≤ x
5          i = i + 1
6          exchange A[i] with A[j]
7  exchange A[i + 1] with A[r]
8  return i + 1
```



[10 marks for a correct answer]

Continued on next page

Question 7

a) Study the adjacency matrix below and draw the undirected graph that it represents.

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

b) Draw the adjacency-list representation of the graph that you have drawn in part (a) above.

c) Using asymptotic notation, how much space (memory) does an adjacency-list representation use to store a graph that has V vertices and E edges?

d) Using asymptotic notation, how much space does an adjacency matrix use to store a graph with V vertices and E edges?

e) Which of the two graph representations, adjacency list or adjacency matrix, is usually more space-efficient for storing a sparse graph? Why?

[2 marks for each part]

Question 8

Suppose that S is a stack, implemented using an array.

a) Write an algorithm in pseudocode that checks to see if the stack, S, is empty. [3 marks]

b) Write an algorithm in pseudocode that pushes a new item, x, onto the stack. [3 marks]

c) Write an algorithm in pseudocode that first checks if S is empty, and then pops an item from S if it is not empty. [4 marks]

Question 9

a) Suppose we have a hash table of size m and that every key, k, that we want to store in the table is a real number less than 1 and greater than or equal to 0. We decide to use the following hash function:

$$h(k) = \lfloor km \rfloor$$

Under what conditions will this hash function work well?

b) Now suppose that the keys, k, to be stored in the hash table are integers greater than or equal to zero (i.e., natural numbers) and we decide to use the division method to calculate the slot indices into the hash table. Write down the formula for the hash function that we would use to do this.

c) Why should m, the size of the hash table, usually not be a power of 2 when we use the division method?

d) What is usually a good choice for the size of the hash table when using the division method?

e) When we use universal hashing, we define a universal class of hash functions, H, from which we randomly select a hash function each time the program is run. Describe one good way in which this universal class of hash functions can be defined.

[2 marks for each correct part]

Continued on next page

Question 10

a) What is the "Liskov substitution principle"? Draw a diagram to illustrate your answer with an example. [4 marks]

b) In the Model-View-Controller (MVC) framework, it is possible to add and change views without changing the model because the model does not know about the views. Which well-known pattern, documented in the Gamma et al. book, is this an example of? [2 marks]

c) In the MVC framework, and many other GUI programming frameworks, it is possible to nest ordinary Views within other Views, called CompositeViews. In this case, should CompositeView be defined to be a superclass of View or should it be defined to be subclass of View? What well-known design pattern is this an example of? [4 marks]

END OF EXAMINATION