Written Examination

Algorithms, Data Structures and Software Engineering for Media Technology

Medialogy 8th semester

27 April 2023, 09.00 – 13.00

Name:____

Cpr.no.: ______________________________________

Study no.: ____________________________________

Aalborg University | MSc in Medialogy | Med8

Algorithms, Data Structures and Software Engineering for Media Technology

Examination

27 April 2023

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.

Question 1

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

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

b) $3n^2\sqrt{n} = \Theta(n^{2.5})$

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

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

e) $\log n = o(1)$

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

g) $\frac{4n^2+3}{2n+9} = \Theta(n^2)$

h) $\sqrt{\frac{n^2 \log n + 4}{n+6}} = \Omega(\sqrt{n})$

i) $2n^3 + 3n = \omega(n^3)$

j) $3\, n^2 \log n = o(n^3)$

[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:

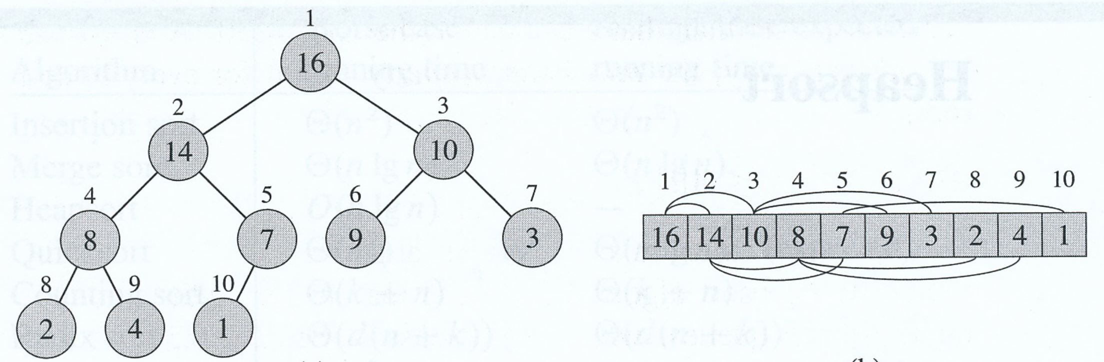
1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for 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) $T(n) = 4T(n/16) + \sqrt{n}$
b) $T(n) = 9T(n/3) + n^3$
c) $T(n) = 8T(n/2) + n^2$
d) $T(n) = 36T(n/6) + 4n^2 + 2n$
e) $T(n) = T(n) + \sqrt{n}$

[2 marks for each correct part]

Question 3



(a) (b)

Figure (a) above shows a binary tree and figure (b) shows a representation of this tree as an array (1-based indexing is used).

a) State the max-heap property. [2 marks]

b) Is the tree in Figure (a) above a max-heap? [2 marks]

c) Suppose *n* is a node in a binary tree and let *i* be the index of node *n* in an array representation of the tree, when the tree is represented in the same manner as that in which the array in Figure (b) above represents the binary tree in Figure (a). Write down algorithms that take *i* as input and efficiently compute the index in the array for
(1) the parent of *n*, [1 mark]
(2) the left child of *n* [1 mark] and
(3) the right child of *n*. [1 mark]

d) For each of the three algorithms you have described in part c), explain how it can be implemented efficiently using bit-shifting. [3 marks]

Question 4

a) Which **one** of the following statements is true?
   A. The POP operation on a stack removes and returns the first element saved in the stack.
   B. The PUSH operation on a stack can lead to stack underflow.
   C. The ENQUEUE operation on a queue can lead to queue overflow.
   D. The DEQUEUE operation on a queue removes and returns the element at the tail of the queue.

b) Which **one** of the following statements is true?
   A. It takes constant time to search for a given key in a doubly-linked list.
   B. It takes linear time to find the minimum key in an unsorted doubly-linked list.
   C. It takes linear time to insert a new element at the head of a doubly-linked list.
   D. It takes quadratic time to delete an element from a doubly-linked list.

c) Which **one** of the following statements is true?
   A. It takes linear time to carry out an inorder tree walk on a binary search tree.
   B. It takes $\Theta(\log_2 n)$ time in the worst case to search for a value in a binary search tree.
   C. It takes $\Theta(n)$ time on average to find the minimum value in a binary search tree.
   D. It takes constant time to insert an element into a binary search tree.

d) Which **one** of the following statements is true?
   A. In the worst case, any comparison sort must make $O(n \log_2 n)$ comparisons.
   B. In the worst case, any comparison sort must make $\Omega(n \log_2 n)$ comparisons.
   C. In the worst case, any comparison sort must make $\omega(n \log_2 n)$ comparisons.
   D. In the worst case, any comparison sort must make $o(n \log_2 n)$ comparisons.

e) Suppose we have an array A of n integers such that the maximum value is *max* and the minimum value is *min* and k = *max* – *min* + 1. Suppose that we have enough memory to allocate an array of size k. Which **one** of the following algorithms can we use to sort A in linear time?
   A. Counting sort
   B. Bucket sort
   C. Merge sort
   D. Quicksort

[2 marks for each correct part]

Question 5

Study the following program and answer the questions that follow it.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define N 3
4
5  __global__ void addVectors(float* dev_a, float* dev_b, float* dev_c) {
6      int i = blockIdx.x;
7      if (i < N) {
8          dev_c[i] = dev_a[i] + dev_b[i];
9      }
10 }
11
12 int main(void) {
13     float a[N] = {1,2,3}, b[N] = {4,5,6}, c[N];
14     float *dev_a, *dev_b, *dev_c;
15
16     cudaMalloc((void**)&dev_a, N*sizeof(float));
17     cudaMalloc((void**)&dev_b, N*sizeof(float));
18     cudaMalloc((void**)&dev_c, N*sizeof(float));
19
20     cudaMemcpy(dev_a, a, N*sizeof(float),cudaMemcpyHostToDevice);
21     cudaMemcpy(dev_b, b, N*sizeof(float),cudaMemcpyHostToDevice);
22
23     addVectors<<<N,1>>>(dev_a, dev_b, dev_c);
24
25     cudaMemcpy(c, dev_c, N*sizeof(float), cudaMemcpyDeviceToHost);
26
27     cudaFree(dev_a);
28     cudaFree(dev_b);
29     cudaFree(dev_c);
30
31     for(int i = 0; i < N; i++)
32         printf("%.0f ", c[i]);
33
34     return EXIT_SUCCESS;
35 }
```

a) What does this program print to the console?
b) In this program, how many thread blocks are there in each grid and how many threads are there in each block?
c) Is the array, a, (defined in line 13) stored in GPU memory or CPU memory?
d) Does the pointer dev_a point at a region of device memory or host memory?
e) In which lines is the GPU memory deallocated?

[2 marks for each correct part]

Question 6

The following pseudocode describes the partition algorithm, as used in quicksort.

```
PARTITION(A, p, r)
x = A[r]
i = p − 1
for j = p to r − 1
    if A[j] ≤ x
        i = i + 1
        exchange A[i] with A[j]
exchange A[i + 1] with A[r]
return i + 1
```

Suppose that the array, *A*, initially contains the following values (note that we are using 1-based indexing – the index of each element in the array is shown above it):

1	2	3	4	5	6	7	8	9	10
4	2	8	7	1	3	5	6	4	9

Suppose that the following call to the PARTITION algorithm is executed:

PARTITION(*A*, 2, 9)

a) Write down the value of each slot in the array *A* after this call to PARTITION has been executed.
b) What value is returned by this call to PARTITION?
c) If we denote the size of *A* by *n*, write down the asymptotic tight upper bound on the running time of PARTITION using appropriate asymptotic order of growth notation (i.e., O, o, Ω, ω or Θ).
d) Write down a loose lower bound on the running time of partition using asymptotic order of growth notation.
e) If the input array *A* is already sorted, what value does PARTITION(*A*, *p*, *r*) return?

[2 marks for each correct part a - e]

Question 7

Suppose you have a data structure, D, that supports the standard dictionary operations, INSERT, SEARCH and DELETE.

a) What is the worst-case running time for the SEARCH operation if D is a hash table and under what conditions does this worst case occur?
b) What is the average-case running time for the SEARCH operation if D is a hash table and under what conditions does this average case occur?
c) What is the worst-case running time of the SEARCH operation if D is a singly-linked list and under what conditions does this worst case occur?
d) If a hash table has m slots and stores n elements, what is its *load factor*?
e) Explain what is meant by the term, *simple unified hashing.*

[2 marks for each correct part]

Question 8

a) Explain the purpose of the *product backlog* in the scrum framework. [2 marks]
b) Explain what a *sprint* is in the scrum framework. [2 marks]
c) Briefly describe the role of each of the following in a scrum team:
    i. the *product owner* [2 marks]
    ii. the *development team* [2 marks]
    iii. the *scrum master* [2 marks]

Question 9

a) For each of the following design patterns, state whether it is *creational*, *structural* or *behavioral*:
   i. Factory method (107)
   ii. Singleton (127)
   iii. Decorator (175)
   [3 marks]
b) In software engineering, explain what is meant by *programming to an interface.* What are some of the advantages of this strategy? [3 marks]
c) Explain the difference between *inheritance* and *composition*. Which of the two is sometimes called "white-box reuse" and why? Which of the two is generally preferable and why? [4 marks]

Question 10

The selection problem is as follows: given an array, A, containing n distinct numbers and an integer, i, such that $1 \leq i \leq n$, find the element x in A that is larger than exactly $i - 1$ other elements in A. We call the result, x, the *ith order statistic* of the set of numbers, A. Describe an algorithm that finds the ith order statistic of a set of n numbers in *expected* linear time. What is the worst-case time of this algorithm?
[10 marks]

END OF EXAMINATION