Carleton University

Final
Examination
Winter 2023

DURATION: 2 HOURS — No. of students: 146

Department Name & Course Number: **Computer Science COMP/MATH 3804**
Course Instructor: Michiel Smid

Authorized memoranda:
Calculator

Students MUST count the number of pages in this examination question paper before beginning to write, and report any discrepancy to the proctor. This question paper has 13 pages (not including the cover page).

This examination question paper MAY be taken from the examination room.

In addition to this question paper, students require:

an examination booklet:	**no**
a Scantron sheet:	**yes**

Instructions:

1. This is a closed book exam.
2. Calculating devices are allowed.
3. All questions must be answered on the scantron sheet.

Marking scheme: Each of the 25 questions is worth 1 mark.

Some useful facts:

1. for any real number $x > 0$, $x = 2^{\log x}$.
2. For any real number $x \neq 1$ and any integer $k \geq 1$,

$$1 + x + x^2 + \cdots + x^{k-1} = \frac{x^k - 1}{x - 1}.$$

3. For any real number $0 < \alpha < 1$,

$$\sum_{i=0}^{\infty} \alpha^i = \frac{1}{1 - \alpha}.$$

Master Theorem:

1. Let $a \geq 1$, $b > 1$, $d \geq 0$, and

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ a \cdot T(n/b) + O\left(n^d\right) & \text{if } n \geq 2. \end{cases}$$

2. If $d > \log_b a$, then $T(n) = O(n^d)$.
3. If $d = \log_b a$, then $T(n) = O(n^d \log n)$.
4. If $d < \log_b a$, then $T(n) = O(n^{\log_b a})$.

1. You are given two recursive algorithms:
   Algorithm A solves a problem of size n by recursively solving 9 subproblems, each of size $n/3$, and performing $\Theta(n^2)$ extra time.
   Algorithm B solves a problem of size n by recursively solving 64 subproblems, each of size $n/8$, and performing $\Theta(n)$ extra time.
   Which of these two algorithms is asymptotically faster?

   (a) Algorithm A.
   (b) Algorithm B.
   (c) Both algorithms have the same running time (up to a constant factor).
   (d) None of the above.

2. Let n be a large integer and consider a set $\{P_1, P_2, \ldots, P_n\}$ of n people. You are given an $n \times n$ matrix K, where

$$K(i,j) = \begin{cases} 0 & \text{if } i = j, \\ 1 & \text{if } i \neq j \text{ and } P_i \text{ knows } P_j, \\ 0 & \text{if } i \neq j \text{ and } P_i \text{ does not know } P_j. \end{cases}$$

   Note that this matrix may not be symmetric. For example, Michiel knows King Charles III, but King Charles III does not know Michiel.
   A *VIP* is a person in this group such that everybody else in the group knows the VIP, whereas the VIP does not know anybody else in the group.
   Is it possible to decide, in $O(n)$ time, if there is a VIP in the group?

   (a) This is possible.
   (b) This is not possible.

3. You are given an infinite array $A[1, 2, \ldots]$ that has the following property: There is an integer $n \geq 2$, such that

   - for all $m = 1, 2, \ldots, n-1$, $A[m] = 0$ and
   - for all $m \geq n$, $A[m] = 1$.

   What is the running time of the fastest algorithm that finds this integer n?

   (a) Since the array has an infinite size, the integer n cannot be found in a finite amount of time.
   (b) $\Theta(n)$
   (c) $\Theta(\log n)$
   (d) $\Theta(n \log n)$

4. You are given a sequence of n numbers, which are not all distinct. You are told that there are only $\sqrt{n}$ many distinct numbers in this sequence.
   What is the running time of the fastest comparison-based algorithm that sorts the entire sequence?

   (a) $\Theta(\log n)$
   (b) $\Theta(n)$
   (c) $\Theta(n \log\log n)$
   (d) $\Theta(n \log n)$

5. The QuickSort algorithm takes as input a sequence S of n distinct numbers. It chooses a pivot p in the sequence S, partitions the sequence into two parts $S_<$, containing all numbers less than p, and $S_>$, containing all numbers larger than p. Then it recursively runs QuickSort twice, once on $S_<$ and once on $S_>$. The running time obviously depends on the pivot p.
   Which of the following is correct?

   (a) QuickSort can be implemented such that the worst-case running time is $O(n \log n)$.
   (b) A worst-case running time of $O(n \log n)$ can only be obtained by choosing the pivot randomly.

6. As you all know, King Charles' coronation will take place on May 6. At the end of the ceremony, Harry and Meghan will run the following randomized algorithm that takes as input an integer $n \geq 1$:

   Algorithm GSTK(n):
   if $n = 1$
   then Harry and Meghan sing "God save the King" once
   else Harry and Meghan sing "God save the King" n times;
   let k be a uniformly random element in $\{1, 2, \ldots, n/2\}$;
   let ℓ be a uniformly random element in $\{n/2, n/2+1, \ldots, n-1\}$;
   GSTK(k);
   GSTK($n - \ell$)
   endif

   What is the expected number of times that Harry and Meghan sing "God save the King" when they run algorithm GSTK(n)?

   (a) $\Theta(\log n)$
   (b) $\Theta(n)$
   (c) $\Theta(n \log n)$
   (d) $\Theta(n^2)$

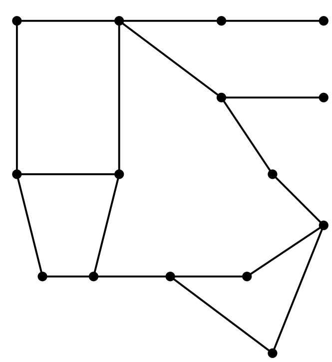
7. Consider a max-heap that stores n pairwise distinct numbers. What is the running time of the fastest algorithm that returns the smallest number in this heap?

   (a) $\Theta(1)$
   (b) $\Theta(\log n)$
   (c) $\Theta(n)$
   (d) $\Theta(n \log n)$

8. Consider a max-heap that stores n pairwise distinct numbers. Professor Uriah Heep has designed a new comparison-based algorithm that supports the operation ExtractMax. What is the largest lower bound on the worst-case running time of this new algorithm?

   (a) $\Omega(1)$
   (b) $\Omega(\log n)$
   (c) $\Omega(n)$
   (d) $\Omega(n \log n)$

9. Is the following graph bipartite?



   (a) The graph is bipartite.
   (b) The graph is not bipartite.

10. Let $G = (V, E)$ be an undirected graph, which is given to you in the adjacency list format. Recall that $degree(u)$ denotes the degree of vertex u. Let $twodegree(u)$ be the sum of the degrees of u's neighbors, i.e.,

$$twodegree(u) = \sum_{v:\{u,v\}\in E} degree(v).$$

What is the running time of the fastest algorithm that computes $twodegree(u)$ for all vertices u?

(a) $\Theta(|V| + |E|)$
(b) $\Theta(|V| \log |V| + |E|)$
(c) $\Theta(|V| + |E| \log |V|)$
(d) $\Theta((|V| + |E|) \log |V|)$

11. Let $G = (V, E)$ be an undirected graph with vertex set $V = \{1, 2, \ldots, n\}$ that is represented by its $n \times n$ adjaceny matrix A.
Consider the square $A^2 = A \cdot A$ of the matrix A.
Let i and j be two distinct vertices of V. What is the meaning of entry (i, j) in the matrix A^2?

(a) The value of this entry is equal to the number of spanning trees of G that contain $\{i, j\}$ as an edge.
(b) The value of this entry is equal to the number of spanning trees of G that do not contain $\{i, j\}$ as an edge.
(c) The value of this entry is equal to the number of paths with at most two edges between the vertices i and j.
(d) The value of this entry is equal to the number of paths with two edges between the vertices i and j.

12. Let $G = (V, E)$ be a directed acyclic graph, let $n = |V|$, let $m = |E|$, and let s and t be two distinct vertices of V. Let $N(s, t)$ denote the number of directed paths in G from s to t. What is the worst-case running time of the fastest algorithm that computes $N(s, t)$?

    (a) $\Theta(m + n \log n)$

    (b) $\Theta(n + m \log n)$

    (c) $\Theta(n + m)$

    (d) Since $N(s, t)$ can be exponential in n, the worst-case running time must be exponential.

13. Let $G = (V, E)$ be a directed graph and, for each edge (u, v) in E, let $\omega(u, v)$ denote its positive weight. For any two vertices x and y of V, we define $\delta(x, y)$ to be the weight of a shortest path in G from x to y.
    We define a new graph $G' = (V, E)$ with the same vertex and edge sets as G. For each edge (u, v) in E, we define its weight in G' to be $\omega'(u, v) = 2023 \cdot \omega(u, v)$. For any two vertices x and y of V, we define $\delta'(x, y)$ to be the weight of a shortest path in G' from x to y.
    Consider the following statement:

    For any two vertices x and y in V, $\delta'(x, y) = 2023 \cdot \delta(x, y)$.

    Which of the following is correct?

    (a) The statement is always true.

    (b) The statement is, in general, false.

14. Let $G = (V, E)$ be a connected undirected graph, in which each edge $\{u, v\}$ has a weight $\omega(u, v)$. Assume that all edge weights are distinct.
    Consider an arbitrary partition of the vertex set V into two disjoint subsets A and B. Let $\{a, b\}$ be the edge in E of minimum weight, with $a \in A$ and $b \in B$.
    Is the following statement true or false?

    $\{a, b\}$ is the only edge in the minimum spanning tree between the sets A and B.

    (a) The statement is always true.

    (b) The statement is, in general, false.

15. Let $G = (V, E)$ be a connected undirected graph, in which each edge $\{u, v\}$ has a weight $\omega(u, v)$. Consider the following algorithm.

    Step 1: Set $E' = E$, $G' = (V, E')$, and $E'' = E$.

    Step 2: While E'' is not empty:

    Step 2.1: Compute an edge e of largest weight in E'', and set $E'' = E'' \setminus \{e\}$.

    Step 2.2: If the graph with vertex set V and edge set $E' \setminus \{e\}$ is connected, then set $E' = E' \setminus \{e\}$ and $G' = (V, E')$.

    Step 3: Return the graph $G' = (V, E')$.

    Which of the following is correct?

    (a) The output is always a minimum spanning tree of the input graph G.
    (b) The output is not necessarily a minimum spanning tree of the input graph G.

16. Let $G = (V, E)$ be a connected undirected graph, in which each edge $\{u, v\}$ has a weight $\omega(u, v)$.
    A *maximum spanning tree* of G is a tree $T = (V, E')$ with $E' \subseteq E$, such that the total weight of all edges in E' is maximized.
    Consider the following algorithm that takes G as input:

    Step 1: Compute a new graph $G' = (V, E)$ with the same vertex and edge sets as G, and in which each edge $\{u, v\}$ has weight $\omega'(u, v) = -\omega(u, v)$.

    Step 2: Compute, and return, a minimum spanning tree of G', using the edge weights ω'.

    Which of the following is correct?

    (a) The output of the algorithm is a maximum spanning tree of G.
    (b) The output of the algorithm is not necessarily a maximum spanning tree of G.
    (c) The algorithm is flawed, because it may not even terminate.
    (d) The algorithm is flawed, because of negative edge weights.

17. Let $G = (V, E)$ be an undirected graph.
An *independent set* in G is a subset X of V such that for any two distinct vertices u and v in X, $\{u, v\}$ is not an edge in E.
A *vertex cover* of G is a subset Y of V such that for every edge $\{u, v\}$, at least one of u and v is in Y.
Is the following statement true or false?

$$X \text{ is an independent set in } G \text{ if and only if } V \setminus X \text{ is a vertex cover of } G.$$

(a) The statement is always true.
(b) The statement is, in general, false.

18. Let $G = (V, E)$ be a directed graph, in which each edge (u, v) has a positive weight $\omega(u, v)$. Let s be a fixed source vertex in V.
Dijkstra's algorithm computes the shortest path lengths from s to all vertices v. It does so by maintaining, for each vertex v, the length of a shortest path from s to v found so far. We modify the algorithm such that it maintains, for each vertex v, the length of a longest path from s to v found so far.

Is the following statement true or false?

The resulting algorithm computes, for each vertex v, the length of a longest path from s to v.

(a) The statement is always true.
(b) The statement is, in general, false.

19. In a dynamic programming algorithm, the solution to the given problem is computed by solving the recurrence bottom-up.

Is the following statement true or false?

By solving the recurrence bottom-up, every dynamic programming algorithm runs in polynomial time.

(a) The statement is true.
(b) The statement is false.

20. In the SubSetSum problem, the input is a tuple $(a_1, a_2, \ldots, a_n, t)$ of positive integers. The question is if there exists a subset I of $\{1, 2, \ldots, n\}$ such that $\sum_{i \in I} a_i = t$.
For each k and ℓ with $1 \leq k \leq n$ and $0 \leq \ell \leq t$, let the Boolean variable $B(k, \ell)$ be True if and only if there exists a subset I of $\{1, 2, \ldots, k\}$ such that $\sum_{i \in I} a_i = \ell$.
For each k and ℓ with $1 \leq k \leq n$ and $\ell < 0$, let the Boolean variable $B(k, \ell)$ be False.

    Which of the following is correct for $2 \leq k \leq n$ and $0 \leq \ell \leq t$?

    (a) $B(k, \ell) =$ True if and only if $B(k-1, \ell) =$ True and $B(k-1, \ell - a_k) =$ True.
    (b) $B(k, \ell) =$ True if and only if $B(k-1, \ell) =$ True and $B(k-1, \ell + a_k) =$ True.
    (c) $B(k, \ell) =$ True if and only if $B(k-1, \ell) =$ True or $B(k-1, \ell - a_k) =$ True.
    (d) $B(k, \ell) =$ True if and only if $B(k-1, \ell) =$ True or $B(k-1, \ell + a_k) =$ True.

21. In the HamiltonCycle problem, we are given an undirected graph G and the question is if there exists a Hamilton cycle in G.
Assume we have a polynomial-time algorithm A that solves the HamiltonCycle problem. Note that this algorithm only returns YES or NO; it does not return anything else.
Let G be a graph that contains a Hamilton cycle.
Which of the following is correct?

    (a) Since HamiltonCycle is **NP**-complete, it is not possible to compute, in polynomial time, a Hamilton cycle in G.
    (b) We can use algorithm A to compute, in polynomial time, a Hamilton cycle in G.

22. Is the following claim true or false?
Let L and L' be two decision problems. Assume that L is in **NP**, $L' \leq_P L$, and L' is **NP**-complete. Then L is also **NP**-complete.

    (a) The claim is true.
    (b) The claim is false.

23. Is the following claim true or false?
If $\mathbf{P} \neq \mathbf{NP}$, then **NP**-complete problems do not exist.

    (a) The claim is true.
    (b) The claim is false.

Definitions for the following two questions:

Let φ be a Boolean formula in the variables $x_1, x_2, \ldots, x_n$. This formula is *satisfiable* if there are truth-values for the variables, such that φ is true.

We say that φ is in *conjunctive normal form* (CNF) if it is of the form

$$\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_m,$$

where each C_i, $1 \leq i \leq m$, is of the form

$$C_i = l_1^i \vee l_2^i \vee \ldots \vee l_{k_i}^i.$$

Each l_j^i is a *literal*, which is either a variable or the negation of a variable.

We say that φ is in *disjunctive normal form* (DNF) if it is of the form

$$\varphi = C_1 \vee C_2 \vee \ldots \vee C_m,$$

where each C_i, $1 \leq i \leq m$, is of the form

$$C_i = l_1^i \wedge l_2^i \wedge \ldots \wedge l_{k_i}^i.$$

Again, each l_j^i is a literal.

We define the following two problems:

$$\text{CNFSAT} = \{\varphi : \varphi \text{ is in CNF-form and is satisfiable}\}$$

and

$$\text{DNFSAT} = \{\varphi : \varphi \text{ is in DNF-form and is satisfiable}\}.$$

24. Which of the following is correct?

    (a) DNFSAT is not in **NP**.

    (b) It is not known if DNFSAT is in **P**.

    (c) DNFSAT is in **P**.

25. Let

$$\text{NON-CNFSAT} = \{\varphi : \varphi \text{ is in CNF-form and is not satisfiable}\}.$$

Is the following statement true or false?
We have seen in class that CNFSAT is in **NP**. By swapping YES and NO in this proof, we obtain a proof that NON-CNFSAT is in **NP**.

   (a) The statement is true.

   (b) The statement is false.