Midterm COMP 3804

March 1, 2023

- All questions must be answered on the scantron sheet.
- Write your name and student number on the scantron sheet.
- You do not have to hand in this examination paper.
- Calculators are allowed.

Marking scheme: Each of the 17 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. The Fibonacci numbers are defined by the recurrence

$$\begin{aligned} F_0 &= 0, \\ F_1 &= 1, \\ F_n &= F_{n-1} + F_{n-2} \text{ if } n \geq 2. \end{aligned}$$

Let A be the matrix

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

Test it manually:

We define A^0 to be the identity matrix, i.e.,

$$n=1: \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$A^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

For $n \geq 1$, A^n denotes the matrix

$$n=2: \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$A^n = \underbrace{A \cdot A \cdots A}_{n \text{ times}}.$$

Is the following true or false? For every integer $n \geq 1$,

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = A^{n-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

$$n=3: \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

(a) True.

(b) False.

2. Consider the recurrence

$$T(n) = T(n/2) + n \log n.$$

Which of the following is true?

(a) $T(n) = \Theta(n)$.

(b) $T(n) = \Theta(n \log n)$.

(c) $T(n) = \Theta(n \log^2 n)$.

(d) None of the above.

$$\begin{aligned} T(n) &= n\log n + T(n/2) \\ &= n\log n + \tfrac{n}{2}\log\tfrac{n}{2} + T(n/4) \\ &= n\log n + \tfrac{n}{2}\log\tfrac{n}{2} + \tfrac{n}{2^2}\log\tfrac{n}{2^2} + T(n/2^3) \\ &\;\;\vdots \\ &= \tfrac{n}{2^0}\log\tfrac{n}{2^0} + \tfrac{n}{2^1}\log\tfrac{n}{2^1} + \ldots + \tfrac{n}{2^{k-1}}\log\tfrac{n}{2^{k-1}} + \underbrace{T(n/2^k)}_{T(1)} \\ &= \sum_{i=0}^{\log n} \tfrac{n}{2^i}\log\tfrac{n}{2^i} \\ &= n\log n \sum \tfrac{1}{2^i} \text{] saw in class } \tfrac{1}{2^i} = 2 \\ &= 2n\log n \\ &= O(n\log n) \end{aligned}$$

3. Consider the recurrence

$$T(n) = n + T(n/17) + T(16n/17).$$

Which of the following is true?

$n=1$: $1/17 + 16/17 = 17/17 = 1$

$\therefore O(n\log n)$

(a) $T(n) = \Theta(n)$.

(b) $T(n) = \Theta(n \log n)$.

(c) $T(n) = \Theta(n \log^2 n)$.

(d) None of the above.

Note:
$=1 \Rightarrow O(n\log n)$
$<1 \Rightarrow O(n)$

4. Consider the following randomized algorithm that takes as input an integer $n \geq 1$:

Algorithm RANDOM(n):
if $n = 1$
then drink one pint of beer
else drink $\boxed{n^2}$ pints of beer;
  let k be a uniformly random element in $\{1, 2, \ldots, n\}$; $n/2$
  RANDOM(k)
endif

What is the expected number of pints of beer that you drink when you run algorithm RANDOM(n)?

Master Theorem:

$T(n) = n^2 + T(n/2)$ $a=1$ $b=2$ $d=2$

$\log_b a = \log_2 1 = 0$

$d>0$

$\hookrightarrow O(n^d)$

$= \Theta(n^2)$

(a) $\Theta(n^3)$

(b) $\Theta(n^2 \log n)$

(c) $\Theta(n^2)$

(d) $\Theta(n)$

5. Consider a sequence of n numbers, where n is a large integer. What is the running time of the fastest comparison-based algorithm that decides if there is a number that occurs at least $n/27$ times in this sequence?

occurences = check the whole thing = $\Theta(n)$

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

(b) $\Theta(\sqrt{n} \log n)$

(c) $\Theta(n \log n)$

(d) $\Theta(n)$

6. Consider the following variant of QuickSort: Given a sequence of n numbers, compute the $(n/3)$-th smallest element, say x, and the $(2n/3)$-th smallest element, say y. Recursively run the algorithm on all numbers less than x, then recursively run the algorithm on all numbers between x and y, and finally, recursively run the algorithm on all numbers larger than y. What is the running time of this sorting algorithm?

run 3 sections

divide into n/3 sections

$\underbrace{3 \cdot T(n/3)}$ + n → $T(n) = 3 \cdot T(n/3) + n$

recurse on sizes n/3

Master Theorem:

a = 3   b = 3   d = 1

$\log_3 3 = 1$ d = 1

↳ Θ(n log n)

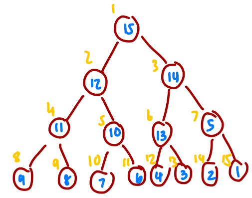
(a) $\Theta(n)$

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

(c) $\Theta(n^2)$

(d) None of the above.

7. Consider a max-heap $A[1 \dots n]$ where $n \geq 15$. Assume that all numbers stored in this max-heap are pairwise distinct. Let x be the fourth largest number stored in this heap. What is the set of indices i such that x may be stored at $A[i]$? ex.



1, 2, ..., 15

x = 12

(a) $\{8, 9, \dots, 15\}$

(b) $\{4, 5, \dots, 15\}$

(c) $\{2, 3, \dots, 15\}$

(d) $\{1, 2, \dots, 15\}$ ⇒ largest num goes in root

8. Consider a max-heap $A[1 \dots n]$ where n is a large integer. Assume that all numbers stored in this max-heap are pairwise distinct. How much time does it take to search for an arbitrary number x in this heap? no "quick" way to search for heaps ⇒ Θ(n)

↳ just search the whole thing

(a) $\Theta(1)$

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

(c) $\Theta(n)$

(d) $\Theta(n \log n)$

9. Consider a max-heap that stores n pairwise distinct numbers. Professor Uriah Heep has developed a new algorithm that supports the operation IncreaseKey in $O(1)$ time. Using this new algorithm, how much time does it take to insert a number into the max-heap?

we know insert() time is O(1) + time for inc.key ⇒ O(1)

(a) $\Theta(1)$

(b) $\Theta(h)$, where h is the height of the node storing the new number in the tree visualizing the heap.

(c) $\Theta(\log n)$

(d) $\Theta(n)$

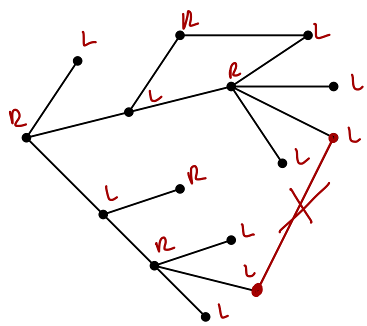
10. You are given two recursive algorithms:
Algorithm A solves a problem of size n by recursively solving 3 subproblems, each of size $n/3$, and performing $\Theta(n^3)$ extra time.
Algorithm B solves a problem of size n by recursively solving 125 subproblems, each of size $n/5$, and performing $\Theta(n^2)$ extra time.
Which of these two algorithms is asymptotically faster?

Algorithm A: $T(n) = 3 \cdot T(n/3) + O(n^3)$
$a=3 \quad b=3 \quad d=3 \qquad \log_3 3 = 1 \quad d>1$
$\hookrightarrow O(n^3)$

" " B: $T(n) = 125 \cdot T(n/5) + O(n^2)$
$a=125 \quad b=5 \quad d=2 \quad \log_5 125 = 3$
$d<3$
$\hookrightarrow O(n^{\log_b a}) = O(n^3)$

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

11. Is the following graph bipartite?



→ odd cycle, so not bipartite

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

12. Let $G = (V, E)$ be an undirected graph. An undirected cycle is a sequence $u_1, u_2, \ldots, u_k$ of pairwise distinct vertices, where $k \geq 3$, such that each of $\{u_1, u_2\}$, $\{u_2, u_3\}$, ..., $\{u_{k-1}, u_k\}$, $\{u_k, u_1\}$ is an edge in E.
Assume the following is given to you: For each connected component of G, you know the number of vertices in this component, and you know the number of edges in this component.
Based on this information only, can you decide if G contains an undirected cycle?

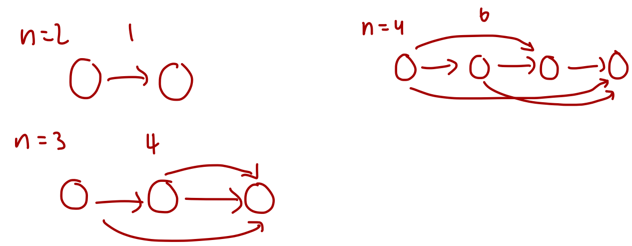
   (a) We can decide if G contains an undirected cycle.
   (b) We cannot decide if G contains an undirected cycle.

13. Let $G = (V, E)$ be a directed graph with $V = \{1, 2, \ldots, n\}$. The adjacency matrix of G is an $n \times n$ binary matrix A, where $A_{ij} = 1$ if and only if the directed edge (i, j) is in E.
Assume you are given the adjacency matrix of G. Is it possible to decide, in $O(n)$ time, if there is a vertex with indegree $n - 1$ and outdegree 0?

IDK lol

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

14. Let $G = (V, E)$ be a directed acyclic graph, let $n = |V|$, 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 largest possible value of $\boxed{N(s, t)}$?



Process of elim.

(a) $\Theta(\cancel{n})$
(b) $\Theta(n \cancel{\log} n)$
(c) $\Theta(\cancel{n^2})$
(d) This number can be exponential in n.

15. Let $G = (V, E)$ be a directed graph that is given using adjacency lists: Each vertex u has a list $\boxed{\text{OUT}(u)}$ storing all edges (u, v) going out of u.
What is the running time of the fastest algorithm that computes, for each vertex v, a list $\boxed{\text{IN}(v)}$ of all edges (u, v) going into v?

we search every adjacency list for every vertex u & record when $\overset{\text{(out)}}{\{u,v\}}$ shows up.

$|V| \to$ every vertex

$|E| \to$ every edge in adj. matrix

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

16. Let $G = (V, E)$ be a directed graph. We run depth-first search on G, i.e, algorithm DFS(G).
Is the following true or false?
The graph G has a directed cycle if and only if the DFS-forest has a cross edge.

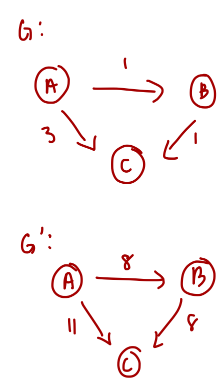
(a) True.
(b) False.

iff it has a backedge

17. Let $G = (V, E)$ be a directed acyclic graph and, for each edge (u, v) in E, let $\text{WT}(u, v)$ 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 $\text{WT}'(u, v) = \text{WT}(u, v) + 7$. 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.
Let x and y be two vertices of V and assume that the shortest path in G from x to y has exactly ℓ edges. Is the following true or false?

$$\delta'(x, y) = \delta(x, y) + 7\ell.$$

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



shortest G:
A→B→C = 2

2+7(2)= 16

shortest G':
A→C = 11

16≠11

Carleton University
Midterm COMP 3804

March 1, 2024

- All questions must be answered on the scantron sheet.
- Write your name and student number on the scantron sheet.
- You do not have to hand in this examination paper.
- Calculators are allowed.

Marking scheme: Each of the 17 questions is worth 1 mark.

Some useful facts:

1. $1 + 2 + 3 + \cdots + n = n(n+1)/2$.
2. for any real number $x > 0$, $x = 2^{\log x}$.
3. For any real number $x \neq 1$ and any integer $k \geq 1$,
$$1 + x + x^2 + \cdots + x^{k-1} = \frac{x^k - 1}{x - 1}.$$
4. 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
$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ a \cdot T(n/b) + \Theta(n^d) & \text{if } n \geq 2. \end{cases}$$
2. If $d > \log_b a$, then $T(n) = \Theta(n^d)$.
3. If $d = \log_b a$, then $T(n) = \Theta(n^d \log n)$.
4. If $d < \log_b a$, then $T(n) = \Theta(n^{\log_b a})$.

1. Recall that $\mathbb{N} = \{1, 2, 3, \ldots\}$ denotes the set of all positive integers. Let $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$ be two functions such that $f(n) = O(g(n))$. Is it true that, for any two such functions f and g,
$$2^{f(n)} = O\left(2^{g(n)}\right)?$$
   (a) This is true.
   (b) This is not true.

2. Consider the recurrence
$$T(n) = \sqrt{n} + T(n/3).$$
   Which of the following is true?

   $T(n) = n^{1/2} + T(n/3)$ $a=1$ $b=3$ $d=1/2$

   $\log_3 1 = 0$ $d>0$

   $\hookrightarrow O(n^{1/2}) = O(\sqrt{n})$

   (a) $T(n) = \Theta(\sqrt{n})$.
   (b) $T(n) = \Theta(\sqrt{n}\log n)$.
   (c) $T(n) = \Theta(n)$.
   (d) $T(n) = \Theta(n \log n)$.

3. Consider the recurrence
$$T(n) = n + T(n/31) + T(29n/31).$$
   Which of the following is true?

   $n=1$: $1/31 + 29/31 = 30/31 < 1 \Rightarrow O(n)$

   (a) $T(n) = \Theta(n)$.
   (b) $T(n) = \Theta(n \log n)$.
   (c) $T(n) = \Theta(n^2)$.
   (d) None of the above.

4. Consider the following recursive algorithm POWER(a, b), which takes as input two integers $a \geq 1$ and $b \geq 1$, and returns a^b:

Algorithm POWER(a, b):
if $b = 1$
then return a
else $c = a^2$;
ANSWER = POWER$(c, \lfloor b/2 \rfloor)$;
if b is even
then return ANSWER
else return $a \cdot$ ANSWER
endif
endif

$T(b) = T(b/2) + O(1)$ $a=1$ $b=2$ $d=0$

$\log_2 1 = 0$ $d=0$

$\hookrightarrow O(\log b)$

Assume that each multiplication, division, and floor-operation in this algorithm takes $O(1)$ time. What is the running time of algorithm POWER(a, b)?

(a) $T(n) = \Theta(\log(a + b))$.
(b) $T(n) = \Theta(\log(ab))$.
(c) $T(n) = \Theta(\log a)$.
(d) $T(n) = \Theta(\log b)$.

5. You are given m sorted arrays $A_1, A_2, \ldots, A_m$, each of length n. Consider the following algorithm that merges these arrays into one single sorted array of length mn:

- $B =$ MERGE(A_1, A_2), where MERGE is the algorithm from class that merges the two sorted arrays A_1 and A_2 into one sorted array B.
- For $i = 3, 4, \ldots, m$, $B =$ MERGE(B, A_i).

$A_1 + A_2 = 2n$ $A_1A_2 + A_3 = 3n$ $\ldots = mn$

$n + 2n + 3n + \ldots + mn$
$(1 + 2 + 3 + \ldots + m)n$
$\hookrightarrow \frac{m(m+1) \cdot n}{2}$

$\frac{m^2n + mn}{2}$

dominant

$O(m^2 n)$

What is the running tims of this algorithm?

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

6. You are given m sorted arrays $A_1, A_2, \ldots, A_m$, each of length n. Assume that m is a power of two. Consider the following algorithm MERGEMANYARRAYS that merges these arrays into one single sorted array of length $\underline{mn}$:

$m = k^2$

Base case: If $m = 1$, then there is nothing to do.

Non-base case: If $m \geq 2$:

- For each $i = 1, 2, \ldots, \boxed{m/2}$, run the MERGE algorithm from class on the two arrays A_{2i-1} and A_{2i}, resulting in a sorted array B_i of length $\boxed{2n}$.
- Recursively run the algorithm MERGEMANYARRAYS on the sorted arrays $B_1, B_2, \ldots, B_{m/2}$.

Let $T(m, n)$ denote the running time of this algorithm. Which of the following is correct?

(a) $T(m, n) = \Theta(mn) + T(m/2, n)$.

(b) $T(m, n) = \Theta(mn) + T(m/2, 2n)$.

(c) $T(m, n) = \Theta(m + n) + T(m/2, n)$.

(d) $T(m, n) = \Theta(m + n) + T(m/2, 2n)$.

$(m/2)\,n = mn/2 = O(mn)$

↳merge

merge time

$T(m/2, 2n)$

merge causes $m/2$ arrays of length $2n$.

7. Professor Uriah Heep has designed a new data structure that stores any sequence of numbers, and supports the following two operations:

- Insert(x): Add the number x to the data structure. This operation takes $\boxed{\Theta(\sqrt{n})}$ time, where n is the current number of elements.
- ExtractMin: Delete, and return, the smallest element stored in the data structure. This operation takes $\boxed{\Theta(\log n)}$ time, where n is the current number of elements.

You use Professor Heep's data structure (and nothing else) to design a sorting algorithm. What is the running time of this sorting algorithm on an input of n numbers?

(a) $\Theta(n \log n)$.

(b) $\Theta(n^{3/2})$.

(c) $\Theta(n^2)$.

(d) None of the above.

Insert elements 1 by 1: $O(\sqrt{1}) + O(\sqrt{2}) + O(\sqrt{3}) + \ldots + O(\sqrt{n})$
$\leq O(\sqrt{n}) + O(\sqrt{n}) + \ldots + O(\sqrt{n}) = O(n\sqrt{n}) = O(n^1 \cdot n^{1/2})$
$= O(n^{2/2} \cdot n^{1/2})$
$= O(n^{3/2})$

extract.min 1 by 1 & dec n: $O(\log n) + O(\log n-1) + \ldots + O(\log 1)$
$\leq O(\log n) + O(\log n) + \ldots + O(\log n) = O(n \log n)$

$\underbrace{O(n^{3/2})}_{\text{dominant term}} + O(n \log n)$

8. Let S be a set of n distinct numbers. Assume this set S is stored in a min-heap $A[1 \ldots n]$. How much time does it take to use this heap to find the largest number of S?

(a) $\Theta(1)$.

(b) $\Theta(\log n)$.

(c) $\Theta(n)$.

(d) $\Theta(n \log n)$.

has to be a leaf

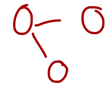
↳look @ all leaves: $O(\lfloor n/2 \rfloor) = O(n)$

9. Let $G = (V, E)$ be a connected undirected graph, and let $n = |V|$. What are the minimum and maximum number of edges that this graph can have?

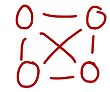
n=1: min=0 O max=0

n=2: min=1 max=1

n=3: min=2 max=3



n=4: min=3 max=6



   (a) 1 ~~and~~ n^2.
   (b) n and $\cancel{n(n-1)/2}$.
   (c) $n-1$ and $\cancel{n^2}$.
   (d) $n-1$ and $n(n-1)/2$.

min: n-1

max n^2: $0^2=0$, $1^2=1$, $\cancel{2^2=4}$

max $n(n-1)/2$: $1(1-1)/2=0$, $2(2-1)/2=1$, $3(3-1)/2=3$, $4(4-1)/2=6$

10. Let $G = (V, E)$ be a directed graph that is given using adjacency lists: Each vertex u has a list $\text{OUT}(u)$ storing all edges (u, v) going out of u.
    What is the running time of the fastest algorithm that computes, for each vertex v, a list $\text{IN}(v)$ of all edges (u, v) going into v?

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

search all vertices adj list's edges & record

$O(|V| + |E|)$

11. Let $G = (V, E)$ be an undirected graph with $n = |V|$ vertices, and assume that the vertex set is stored in an array $V[1 \dots n]$. For each i, let $v_i = V[i]$.
    Is it possible to give each edge $\{v_i, v_j\}$ a direction (i.e., replace it by exactly one of (v_i, v_j) and (v_j, v_i)) such that the resulting directed graph is acyclic?

IDK how

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

12. Let $G = (V, E)$ be an undirected graph, and assume that this graph is stored using the adjacency matrix. What is the running time of the fastest depth-first search algorithm for this graph?

comeback to this

   (a) $\Theta(|V| + |E|)$.
   (b) $\Theta(|V|^2 + |E|^2)$.
   (c) $\Theta(|V|^2)$.
   (d) $\Theta(|E|^2)$.

13. Let $G = (V, E)$ be a directed acyclic graph, and let s and t be two distinct vertices of V. What is the running time of the fastest algorithm that computes the number of directed paths in G from s to t?

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

14. Let $G = (V, E)$ be a directed graph. We run depth-first search on G, i.e, algorithm DFS(G). Recall that this classifies each edge of E as a tree edge, forward edge, back edge, or cross edge.
    Let (u, v) be an edge of E that is not classified as a tree edge.
    Is the following true or false?
    It is possible to run algorithm DFS(G), where vertices and edges are processed in a different order, such that (u, v) is classified as a tree edge.

    (a) True.
    (b) False.

15. Let $G = (V, E)$ be a directed graph. We run depth-first search on G, i.e, algorithm DFS(G). Is the following true or false?
    If the graph G has a directed cycle that contains a forward edge, then G also contains a directed cycle that does not contain a forward edge.

    (a) True.
    (b) False.

16. Let $G = (V, E)$ be a directed acyclic graph and, for each edge (u, v) in E, let $\textsc{wt}(u, v)$ denote its positive weight. Let s be a source vertex, and for each vertex v, let $\delta_{\max}(s, v)$ be the weight of a *longest* path in G from s to v.
    What is the running time of the fastest algorithm that computes $\delta_{\max}(s, v)$ for all vertices v?

    (a) Since there can be exponentially many paths from s to some vertex v, the running time must be at least exponential.
    (b) $\Theta((|V| + |E|) \log |V|)$.
    (c) $\Theta(|E| + |V| \log |V|)$.
    (d) $\Theta(|V| + |E|)$.

17. After this midterm, you go to a Karaoke Bar and sing the following randomized and recursive song AWESOMEST(n), which takes as input an integer $n \geq 1$:

> **Algorithm** AWESOMEST(n):
> sing the following line n times:
> *COMP 3804 is the awesomest course I have ever taken*;
> **if** $n \geq 2$
> **then** let k be a uniformly random element in $\{1, 2, \ldots, n\}$;
>   AWESOMEST(k)
> **endif**

$n + T(n/2)$

$a=1 \quad b=2 \quad d=1$

$\log_2 1 = 0 \quad d>0$

$O(n)$

What is the expected number of times you sing *COMP 3804 is the awesomest course I have ever taken*?

(a) $\Theta(n)$.

(b) $\Theta(n \log n)$.

(c) $\Theta(n^2)$.

(d) None of the above.