Midterm Solution

COMPSCI 220: WEEK 13.3

Instructor: Meng-Fen Chiang

THE UNIVERSITY OF AUCKLAND
Te Whare Wananga o Tamaki Makaurau
NEW ZEALAND

Limit Rule

Suppose that $L := \lim_{n\to\infty} f(n)/g(n)$ exists. Then,

- if $L = 0$ then f is $O(g)$ and f is not $\Omega(g)$;
- if $0 < L < \infty$ then f is $\Theta(g)$;
- if $L = \infty$ then f is $\Omega(g)$ and f is not $O(g)$.

- When f and g are **positive** and **differentiable** functions for $n > 0$, one of the following satisfies:
  - $\lim_{n\to\infty} f(n) = \infty$ and $\lim_{n\to\infty} g(n) = \infty$
  - $\lim_{n\to\infty} f(n) = 0$ and $\lim_{n\to\infty} g(n) = 0$

  L'Hopital rule of calculus can be applied:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f'(n)}{g'(n)}$$

Question 1 (A)

(A) Prove that $T(n) = n^4 + 2n^3 + 3n^2 + 10n$ is both $O(n^4)$ and $O(n^5)$.

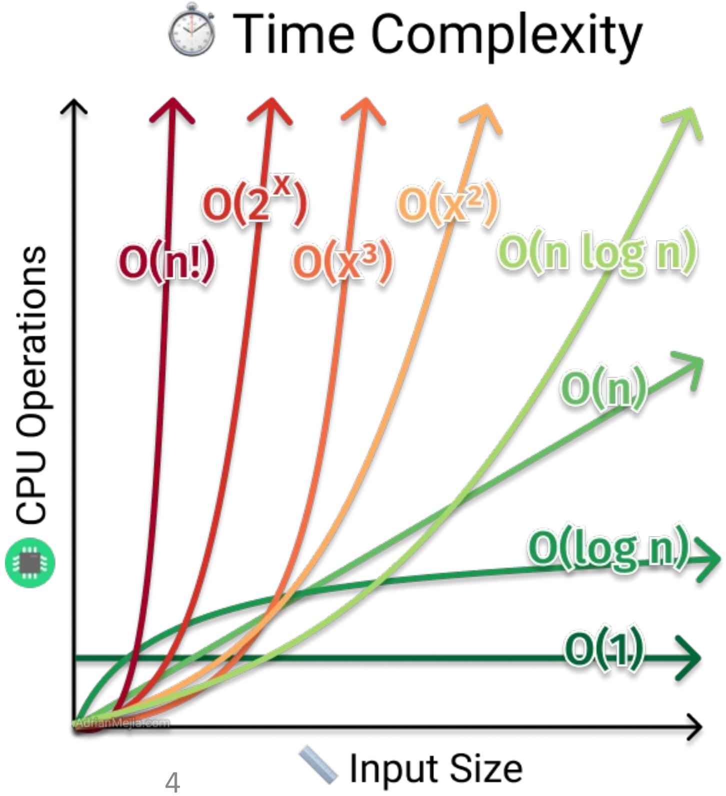
$\lim_{n\to\infty} \frac{n^4+2n^3+3n^2+10n}{n^4} = 3$. This means $T(n)$ is $\Theta(n^4)$, and is also $O(n^4)$.

$\lim_{n\to\infty} \frac{n^4+2n^3+3n^2+10n}{n^5} = 0$. This means $T(n)$ is $O(n^5)$.

Question 1 (B)

(B) Consider the following functions of n: $2n^2, n\lg_3^n, 0.1n^{3/2}, n!, 2^n$. Put them in order from smallest to largest asymptotic growth rate. **[8 marks]**

$$n\lg_3^n < 0.1n^{3/2} < 2n^2 < 2^n < n!$$



Question 1 (C)

(C) Let $T(n) = n^{1/50} + 5n^{1/100}\log_2 n + (\log_2 n)^2$ be processing time of an algorithm for input of size n. Which is the asymptotic time complexity of this algorithm, $\Theta(n^{1/50})$ or $\Theta((\log n)^2)$? Please show your working to justify your answer. **[8 marks]**

$T(n) = n^{1/50} + 5n^{1/100}\log_2 n + (\log_2 n)^2$. The dominant term is $n^{1/50}$, because $\log_2 n < n^k$ where $k > 0$. We can show this using limit rule and L'Hopital's rule:

$$
\begin{aligned}
\lim_{n\to\infty} \frac{\log_2 n}{n^k} &= \lim_{n\to\infty} \frac{\ln n}{\ln 2 \cdot n^k} && \text{(Change the base of logarithm)} \\
&= \lim_{n\to\infty} \frac{1}{\ln 2 \cdot n \cdot k n^{k-1}} && \text{(Apply L'Hopital's rule)} \\
&= \lim_{n\to\infty} \frac{1}{\ln 2 \cdot k n^k} && \text{(Because} k > 0, n^k \to \infty) \\
&= 0
\end{aligned}
$$

As such, the asymptotic time complexity is $n^{1/50}$.

Question 2.1

(2.1) Which of the following statements is **TRUE**? **[5 marks]**

(A) Insertion sort is preferred to the other sorting algorithms when the input arrays are almost sorted.

(B) Insertion sort is never preferred to the more sophisticated algorithm.

(C) Merge sort best complexity of $\Theta(n)$.

(D) Insertion sort has worst complexity of $\Theta(n \log n)$.

(E) Insertion sort should always be preferred to Merge sort.

Question 2.2

(2.2) Which of the following statements is **TRUE** about the quicksort algorithm? [**5 marks**]

(A) The average time of Quicksort is $\Theta(n \log(n))$.

(B) Quicksort is stable.

(C) Quicksort is in-place.

(D) The best, worst and average time of Quicksort is $\Theta(n \log(n))$.

(E) None of the above.

Question 3

- Determine the order of the list after partitioning [41, 29, -100, 20, 15, 77, 10], assume the pivot is 20.

- Step1: First, we swap the pivot with the first element in the list.

[<u>20</u>, 29, -100, 41, 15, 77, 10]

- Step2: Next, we have the two pointers L and R starting on each end of the list and looks for elements bigger than the pivot and smaller than the pivot respectively. L pointer will find 29 and R pointer will find 10 for the first time. Swap 29 and 10.

[<u>20</u>, 10, -100, 41, 15, 77, 29]

Question 3 (Contd.)

- Step3: Continue to move L and R will lead to L finding 41 and R finding 15. Swap 41 and 15.

[20, 10, -100, 15, 41, 77, 29]

- Step4: Now, when the R moves to the left again, it will collide with L. This is when we swap this indexed element with the pivot.

[15, 10, -100, 20, 41, 77, 29]

- At this point, all elements smaller than the pivot are on its left and all elements larger than the pivot are on its right. The first partition is done.

Question 4

We run the array implementation of the heapsort algorithm. We have built the binary heap [50,13,9,8,7,6,5,4,3,1]. Which of the following arrays corresponds to the next step of the algorithm?

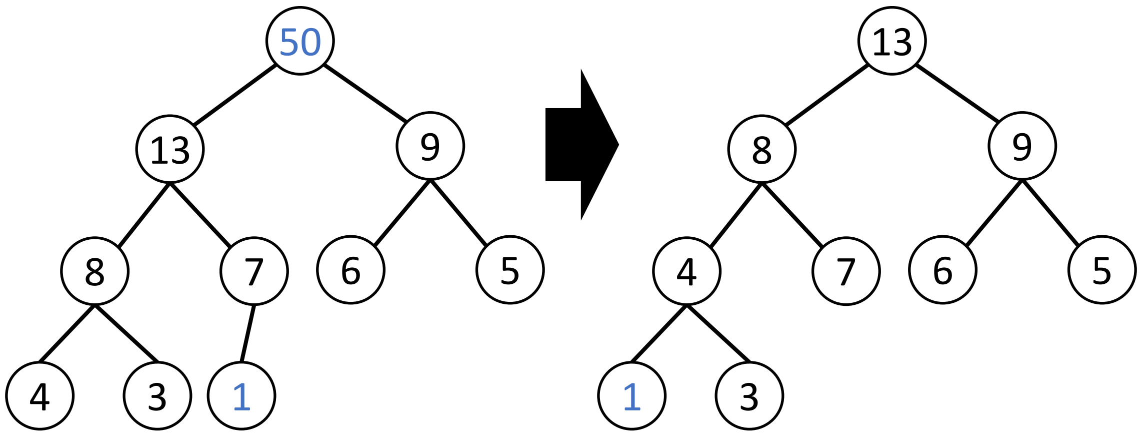
(A) Remove 50 and get [13,9,7,8,4,6,5,1,50].

(B) Remove 50 and get [13,9,7,8,3,6,5,4,1,50].

(C) Remove 4 and get [50,13,9,8,7,6,5,3,1].

(D) Remove 50 and get [13,8,9,4,7,6,5,1,3,50].

(E) Remove 1 and get [50,13,9,8,7,6,5,4,3].



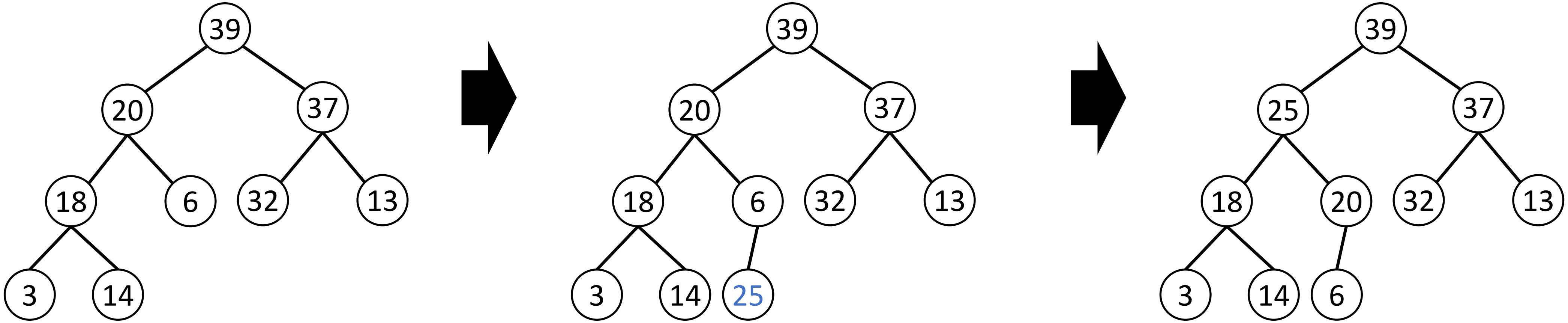
Question 5 (A)

Consider the following maximum heap: [39, 20, 37, 18, 6, 32, 13, 3, 14].

(A) Insert 25 to the heap.

(B) Delete 39 from the heap derived from previous step.

Insert 25 to the heap. [39,25,37,18,20,32,13,3,14,6]



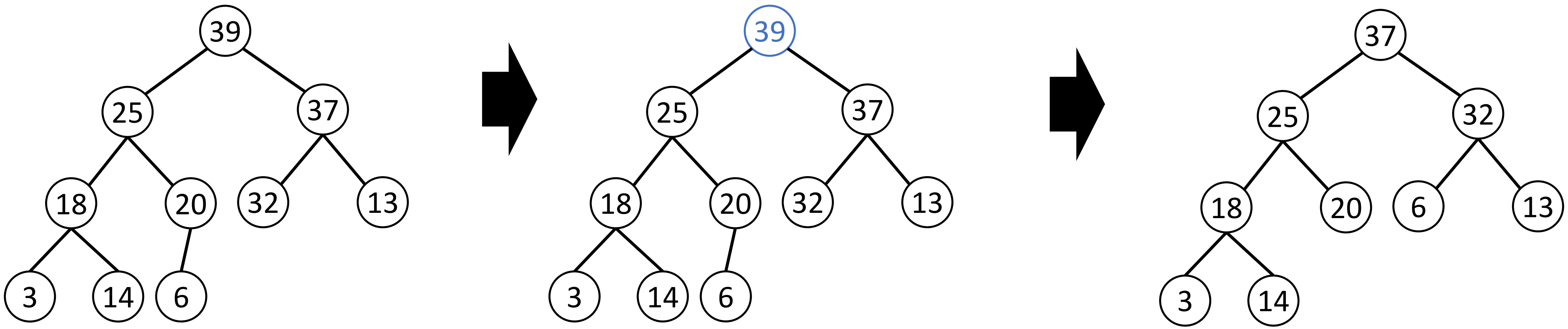
Question 5 (B)

Consider the following maximum heap: [39, 20, 37, 18, 6, 32, 13, 3, 14].

(A) Insert 25 to the heap.

(B) Delete 39 from the heap derived from previous step.

Delete 39 from the heap derived from previous step. [37,25,32,18,20,6,13,3,14]



Question 6

We are looking for 9 in [0,1,3,4,13,19,-100]. Which of the following statements best describes the first steps of a binary search algorithm?

(A) Find the middle element. It is 3. Because $9 > 3$ then look for 9 in [4,13,19,-100].

(B) Find the middle element. It is 3. Because $9 > 3$ then look for 9 in [0,1].

(C) Find the middle element. It is 3. Because $9 > 3$ then look for 9 in [3,4,13,19,-100].

(D) Take 0 as the pivot. Partition the list into [] and [1,3,4,13,19]. Because $9 > 0$ we recursively search for 9 in [1,3,4,13,19,-100].

(E) We cannot use binary search because the list is not sorted.

Question 7 (A)

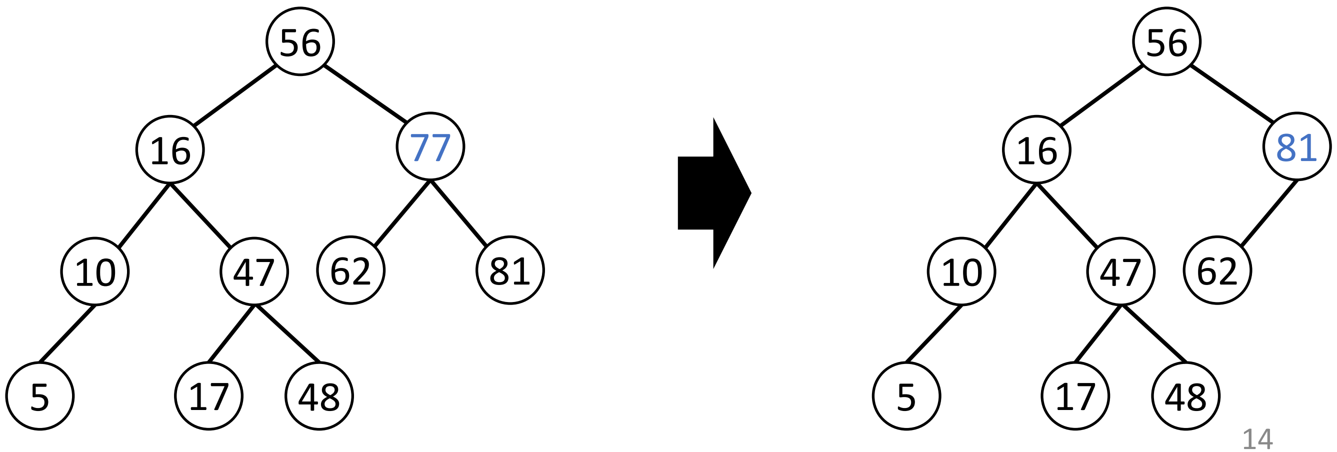
Describe the process and the outcome of the following deletion operations on τ.

(A) Delete node 77 in the tree τ by using the minimum key in the right subtree.

(B) Delete node 16 in the tree τ by using the minimum key in the right subtree.

(C) Delete node 16 in the tree τ by using the maximum key in the left subtree.

Node 77 has two children: Find the minimum key K = 81 in the right subtree, delete that node, and replace the key of node 77 by K.



Question 7 (B)

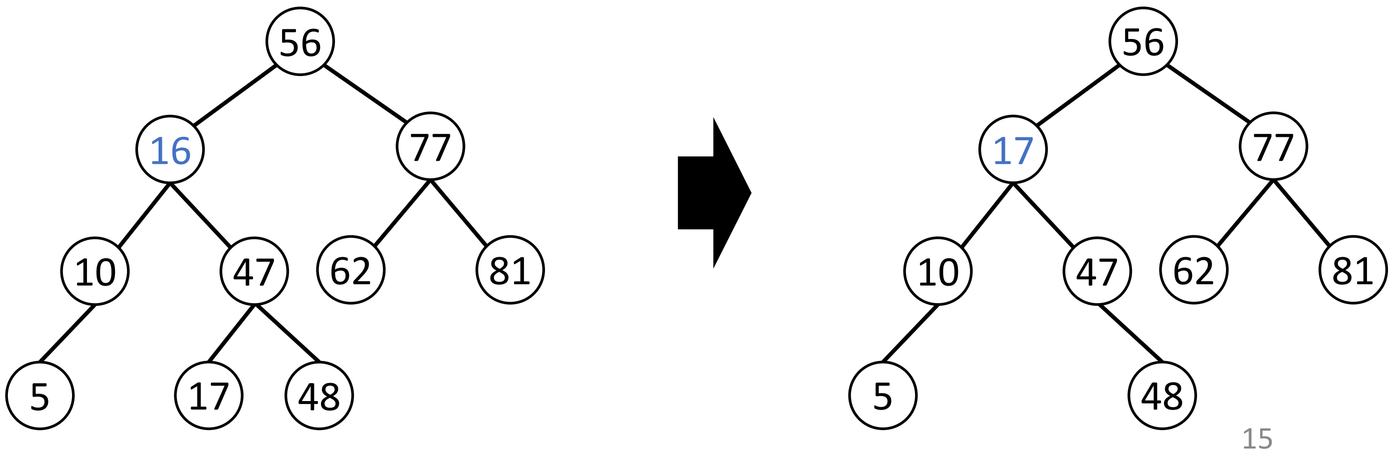
Describe the process and the outcome of the following deletion operations on τ.

(A) Delete node 77 in the tree τ by using the minimum key in the right subtree.

(B) Delete node 16 in the tree τ by using the minimum key in the right subtree.

(C) Delete node 16 in the tree τ by using the maximum key in the left subtree.

Node 16 has two children: Find the minimum key K = 17 in the right subtree, delete that node, and replace the key of node 16 by K.



Question 7 (C)

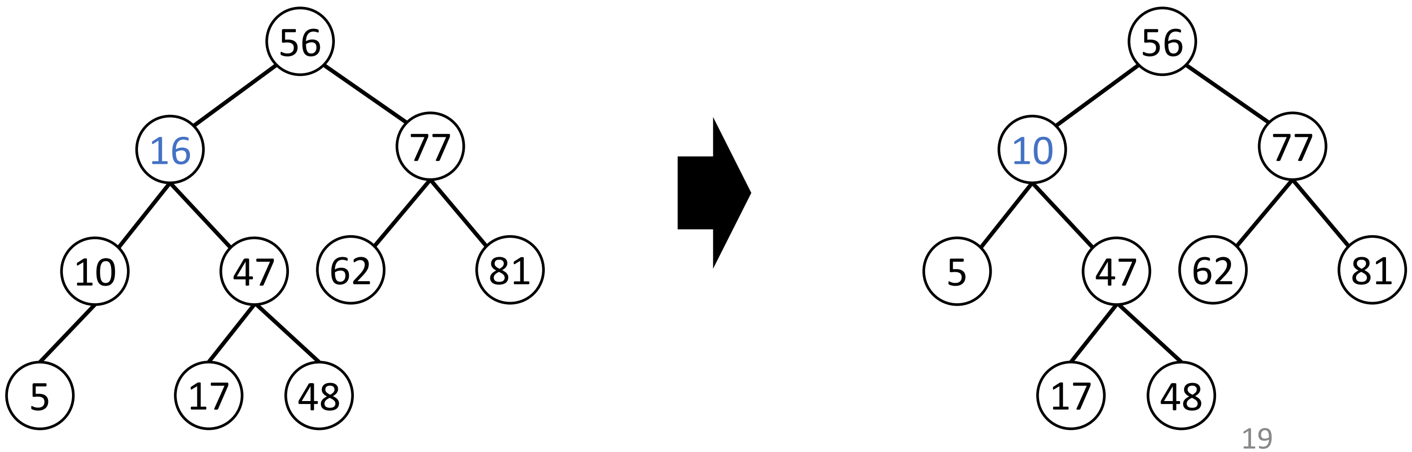
Describe the process and the outcome of the following deletion operations on τ.

(A) Delete node 77 in the tree τ by using the minimum key in the right subtree.

(B) Delete node 16 in the tree τ by using the minimum key in the right subtree.

(C) Delete node 16 in the tree τ by using the maximum key in the left subtree.

Node 16 has two children: Find the maximum key K = 10 in the left subtree, delete that node, and replace the key of node 16 by K.

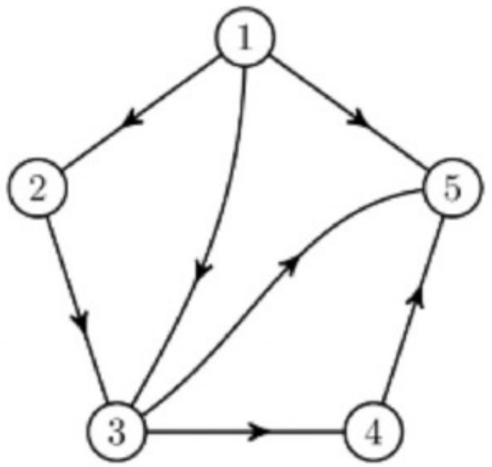


Question 8

(A) What is the source node and sink node of G?

(B) What is the adjacency list of this digraph G?

(C) What is the adjacency matrix of this digraph?



(A)

source = 1
sink = 5

(B)

Adjacency list

1 : 2, 3, 5
2 : 3
3 : 4, 5
4 : 5
5 :

(C)

Adjacency matrix $\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

Question 9

Which of the following statements about graph data operation is **TRUE**?

(A) In graph adjacency list, it takes $\Theta(n)$ to compute the in-degree of a vertex.

(B) It takes less complexity to delete a vertex from the adjacency matrix than from the adjacency list for sparse graphs.

(C) In graph adjacency matrix, it takes $\Theta(n)$ to compute the out-degree of a vertex.

(D) To add a vertex, it takes $\Theta(n)$ for both adjacency matrix and adjacency list representations.

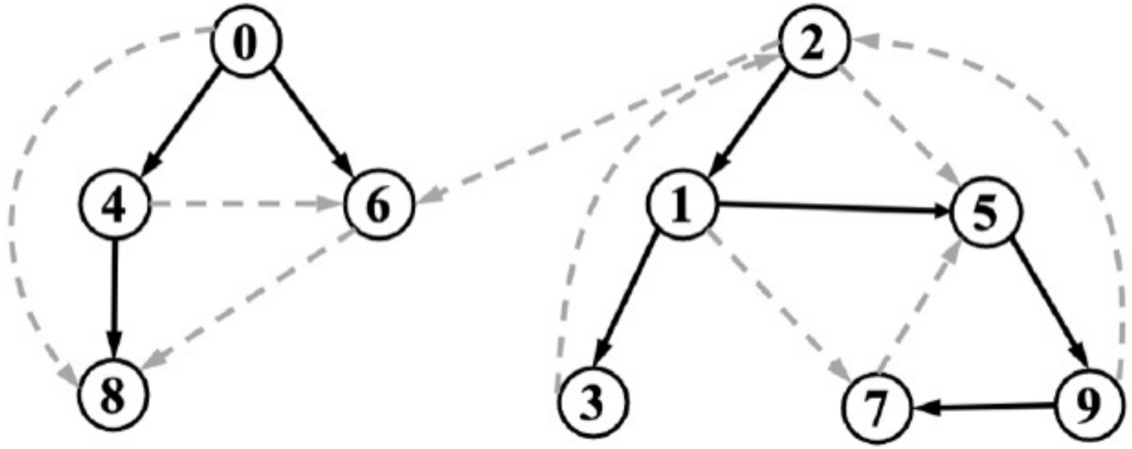
(E) In graph adjacency list, it takes $\Theta(n)$ to compute the out-degree of a vertex.

Adjacency Lists / Matrices: Comparative Performance

Operation	array/array	list/list
arc (i, j) exists?	$\Theta(1)$	$\Theta(d)$*
out-degree of i	$\Theta(n)$	$\Theta(1)$
in-degree of i	$\Theta(n)$	$\Theta(n + e)$
add arc (i, j)	$\Theta(1)$	$\Theta(1)$
delete (i, j)	$\Theta(1)$	$\Theta(d)$
add node	$\Theta(n)$	$\Theta(1)$
delete node i	$\Theta(n^2)$	$\Theta(n + e)$

Question 10

Consider the search forest of a digraph G after running the general graph traversal algorithm. Which of the following statement is **TRUE**?



(A) Node 5 is coloured grey before node 2.

(B) Node 0 is coloured grey before node 2.

(C) Arc (6,8) is a forward arc.

(D) Arc (1,7) is a cross arc.

(E) None of the other answers are correct.