In this exam:

- The height of a singleton tree is defined as 0.
- The order of a node in a tree is defined as the number of its children.
- The order of a tree is defined as the maximum order among its nodes.
- Unless otherwise specified, all logarithms are taken with respect to base 2, i.e., $\log_2$.

一、(20%)是非題（每題 2 分，答錯每題倒扣 1 分至本大題 0 分止）

1. Given a connected graph $G = (V, E)$, if a vertex $v \in V$ is visited during level k of a breadth-first search from source vertex $s \in V$, then every path from s to v has length at most k.
2. Depth-first search will take $\Theta(V^2)$ time on a graph $G = (V, E)$ represented as an adjacency matrix.
3. In a weighted undirected graph $G = (V, E, w)$, breadth-first search from a vertex s finds single-source shortest paths from s (via parent pointers) in $O(V + E)$ time.
4. In a weighted undirected **tree** $G = (V, E, w)$, breadth-first search from a vertex s finds single-source shortest paths from s (via parent pointers) in $O(V + E)$ time.
5. In a weighted undirected **tree** $G = (V, E, w)$, **depth-**first search from a vertex s finds single-source shortest paths from s (via parent pointers) in $O(V + E)$ time.
6. Dijkstra's shortest-path algorithm may relax an edge more than once in a graph with a cycle.
7. Given a weighted directed graph $G = (V, E, w)$ and a source $s \in V$, if G has a negative-weight cycle somewhere, then the Bellman-Ford algorithm will always compute an incorrect result for some $\delta(s, v)$.
8. In a weighted directed graph $G = (V, E, w)$ containing no zero- or positive-weight cycles, Bellman-Ford can find a longest (maximum-weight) path from vertex s to vertex t.
9. In a weighted directed graph $G = (V, E, w)$ containing a negative-weight cycle, running the Bellman-Ford algorithm from s will find a shortest acyclic path from s to a given destination vertex t.
10. Given a weighted directed graph $G = (V, E, w)$ and a shortest path p from s to t, if we doubled the weight of every edge to produce $G' = (V, E, w')$, then p is also a shortest path in G'.

二、(64%)單選題（每題 2 分，答錯不倒扣）

Please choose the best possible answer in each of the following questions. In questions that ask for time complexities, please choose the tightest bound.

11. In a binary heap of size n, the worst case time complexity of one insert operation is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)
12. In a binary heap of size n, the worst case time complexity of one find-min operation is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)
13. In a binary heap of size n, the worst case time complexity of one decrease-key operation is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)
14. In a binomial heap of size n, the worst case time complexity of $O(\log n)$ consecutive insert operations is
(A) $O(\log n)$ (B) $O(\log^2 n)$ (C) O(n log n) (D) $O(n \log^2 n)$ (E) $O(n^2 \log n)$
15. In a binomial heap of size n, the worst case time complexity of one delete-min operation is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)

16. The worst case time complexity of one union operation upon two binomial heaps of total size n is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)
17. In a binomial heap of size n, the worst case time complexity of one decrease-key operation is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)
18. In a Fibonacci heap of size n, the worst case time complexity of one find-min operations is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)
19. In a Fibonacci heap of size n, the worst case time complexity of one delete-min operations is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)
20. The worst case time complexity of one union operation upon two Fibonacci heaps of total size n is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)
21. In a leftist heap of size n, the worst case time complexity of one delete-min operation is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)
22. The worst case time complexity of one merge operation upon two leftist heaps of total size n is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)
23. In a leftist heap of size n, the worst case time complexity of one decrease-key operation is
(A) O(1) (B) O(log*n) (C) O(log n) (D) O(n) (E) O(n log n)
24. An AVL tree of 100 nodes has height at least
(A) 5 (B) 6 (C) 7 (D) 8 (E) 9
25. A 2-3-4 tree of 255 nodes has height at least
(A) 3 (B) 4 (C) 5 (D) 6 (E) 7
26. How many binomial trees are there in a binomial heap with 2020 items?
(A) 1 (B) 3 (C) 5 (D) 7 (E) 9
27. How many items are there in a binomial tree with order 4?
(A) 13 (B) 14 (C) 15 (D) 16 (E) 17
28. Which of the following is correct?

(A) $\frac{n^2}{\log n} = O(n^{1.5})$

(B) $2^{2n} = \Theta(2^n)$

(C) If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$

(D) If $f(n) \neq O(g(n))$, then $g(n) = O(f(n))$

29. Order the following functions by the big-Oh notation in decreasing order.
(*a*) 1.001^n (*b*) $n!$ (*c*) $n^{5/2}$ (*d*) $2^{7\log n}$ (*e*) 3^{100000} (*f*) $\log\log n$ (*g*) $\sqrt{n}$ (*h*) n^n
Which of the following order is correct?
(A) $h > a > b > d > c > g > f > e$
(B) $h > b > a > d > c > g > f > e$
(C) $h > b > a > d > c > f > g > e$
(D) $b > h > a > d > c > g > f > e$

30. If $T(n) = 2T(\sqrt{n}) + \log n$, where $T(2) = 0$, then $T(n) =$?
(A) $n/\log n$
(B) $\log n\,(\log\log n)$
(C) $\log\log n$
(D) $\log n$

31. Which is the tight lower bound for any comparison based sorting algorithms?
(A) $\Omega(n)$
(B) $\Omega(n^2)$
(C) $\Omega(n \log n \log n)$
(D) $\Omega(n \log n)$

接次頁

32. What's the time complexity to do (deterministic) quick sort, insertion sort and merge sort on a list of numbers in perfect sorted order?
    (A) quick sort: $O(n^2)$, insertion sort: $O(n)$, merge sort: $O(n\log n)$
    (B) quick sort: $O(n\log n)$, insertion sort: $O(n^2)$, merge sort: $O(n\log n)$
    (C) quick sort: $O(n)$, insertion sort: $O(n^2)$, merge sort: $O(n\log n)$
    (D) quick sort: $O(n^2)$, insertion sort: $O(n^2)$, merge sort: $O(n)$

33. To search a specific number in a list, there are following two strategies:
    (a) Linear Search
    (b) First sort the list using merge sort, and then do binary search
    Which strategy is faster in average? What's its corresponding time complexity?
    (A) (a), $O(n)$
    (B) (b), $O(\log n)$
    (C) (a), $O(\log n)$
    (D) (b), $O(n\log n)$

34. Consider the following list of 9 activities with their start time s and end time t, what's the maximum number of activities that can be performed by a single person? Assuming that a person can only work on a single activity at a time.

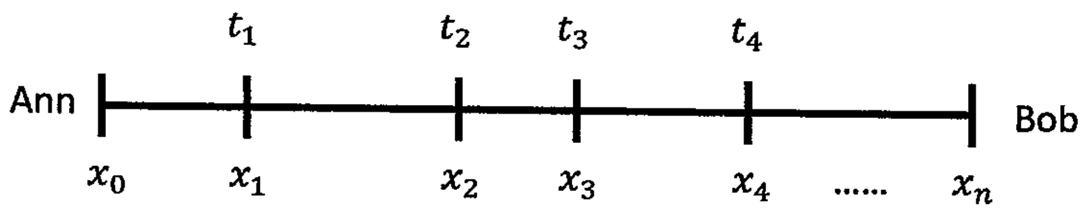
index	1	2	3	4	5	6	7	8	9
s	2	3	1	10	14	6	13	7	19
t	5	9	12	16	17	15	18	11	20

    (A) 2
    (B) 3
    (C) 4
    (D) 5

35. Consider the following two sequences:
    3, 2, 1, 7, 9, 8, 6, 8, 0, 9
    8, 3, 7, 8, 1, 9, 8, 3, 9, 0
    The length of the longest common subsequence is:
    (A) 3
    (B) 5
    (C) 4
    (D) 2

見背面

For problems 36 to 42, please refer to the following dynamic programming problem.



Ann is riding scooter from her home to Bob's home with a constant speed v. However, her scooter can only hold gas for 100kms riding at most. There are a few gas stations along the way at $x_1, x_2, \dots, x_n$ distances from her home, but filling up at different stations x_i would take different time t_i . She would start her trip with a full tank of gas, and if she decides to fill up at a gas station, she needs to fill her entire tank up. Since the riding time is constant (x_n/v), we only need to consider the time consumed by filling gas. (Assume that the distances between two consecutive stations are not longer than 100kms. Bob's home is at x_n distances from Ann's home and the distances between his home and the last station are not longer than 100kms as well.)

36. If $s[a]$ indicates the minimum time she has spent on filling gas when she has filled the gas at a^{th} station, then

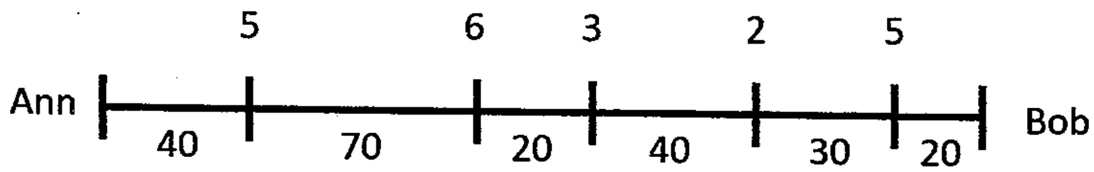
(A) $s[a] = \begin{cases} 0, & \text{if } a = 0 \\ \min_{(0 \le i \le a-1) \wedge (x_a - x_i \le 100)} (s[i] + t[i]), & \text{otherwise} \end{cases}$

(B) $s[a] = \begin{cases} 0, & \text{if } a = 0 \\ \min_{(0 \le i \le a-1) \wedge (x_a - x_i \le 100)} (s[i] + t[a]), & \text{otherwise} \end{cases}$

37. If $r[a]$ indicates the minimum time she has spent on filling gas when she arrives at a^{th} gas station, then

(A) $r[a] = \begin{cases} 0, & \text{if } a = 0 \\ \min_{(0 \le i \le a-1) \wedge (x_a - x_i \le 100)} (r[i] + t[i]), & \text{otherwise} \end{cases}$

(B) $r[a] = \begin{cases} 0, & \text{if } a = 0 \\ \min_{(0 \le i \le a-1) \wedge (x_a - x_i \le 100)} (r[i] + t[a]), & \text{otherwise} \end{cases}$



According to the figure, solve $s[1], \dots, s[5]$.

38. $s[1] =$
 (A) 0
 (B) 5
 (C) 6
 (D) 40

39. $s[2] =$
 (A) 0
 (B) 5
 (C) 6
 (D) 11

40. $s[3] =$
(A) 5
(B) 8
(C) 11
(D) 13

41. $s[4] =$
(A) 8
(B) 10
(C) 11
(D) 13

42. $s[5] =$
(A) 8
(B) 10
(C) 11
(D) 13

三、(16%) 複選題（每題 2 分，每答錯一個選項倒扣 0.4 分至該題 0 分止）

43. Consider the expression tree in Fig.1
(A) The prefix expression is a+b*c+d+e*f*g*h+i+j*k
(B) The prefix expression is *++a*bc*+def*g++hi*jk
(C) The infix expression is *+*+*g+a*+f+*bcdehijk
(D) The postfix expression is abc*+de+f*+ghi+jk*+**
(E) The postfix expression is k*j+i+h*g*f*e+d+c*b+a

44. In a splay tree of size n
(A) The height is $O(\log n)$.
(B) Time complexity of one insertion is $O(\log n)$.
(C) Time complexity of n insertions is $O(n \log n)$.
(D) Time complexity of one deletion is $O(\log n)$.
(E) Time complexity of n deletions is $O(n \log n)$.

45. Consider the splay tree in Fig.2.
(A) In Fig.2, after inserting the key 93, then the parent of key 83 is key 79.
(B) In Fig.2, after inserting the key 80, then the parent of key 80 is key 79.
(C) In Fig.2, after inserting the key 33, then the parent of key 35 is key 48.
(D) In Fig.2, after inserting the key 1, then the parent of key 15 is key 2.
(E) In Fig.2, after inserting the key 40, then the root is key 40.

46. In a red-black tree of size n
(A) Time complexity of one search is $O(\log n)$.
(B) Time complexity of one insertion is $O(\log n)$.
(C) Time complexity of n insertions is $O(n \log n)$.
(D) Time complexity of one deletion is $O(\log n)$.
(E) Time complexity of n deletions is $O(n \log n)$.

47. Which of the followings are balanced trees?
(A) Binary search tree.
(B) AVL tree.
(C) Red-black tree.
(D) Splay tree.
(E) 2-3-4 tree.

見背面

48. In an AA tree, which of the following statements are correct?
(A) There are no two consecutive right horizontal links.
(B) There is no left horizontal link.
(C) A node has the same level as its right horizontal link child.
(D) A node has the same level as its left-link child.
(E) The height is $O(\log n)$.

49. Which of the following operations are supported by binary search trees?
(A) Find key
(B) Insert key
(C) Delete key
(D) Find-Min
(E) Delete-Min

50. Consider the Fibonacci heap in Fig.3
(A) In Fig.3, after decreasing key 89 to 40, there will be 3 marked nodes in total.
(B) In Fig.3, after decreasing key 59 to 40, there will be 5 marked nodes in total.
(C) In Fig.3, after decreasing key 15 to 10, there will be 2 trees in total.
(D) In Fig.3, after delete-min, there will be two trees.
(E) In Fig.3, after inserting key 68, there will be one tree.

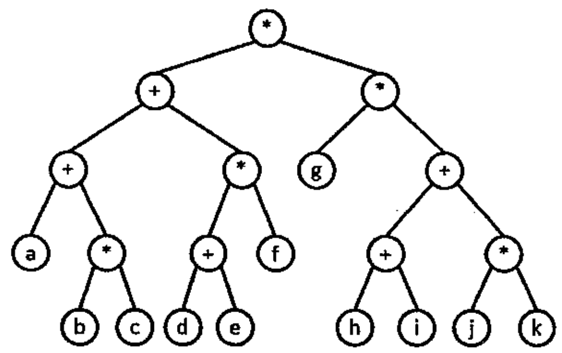


Fig.1 Expression tree

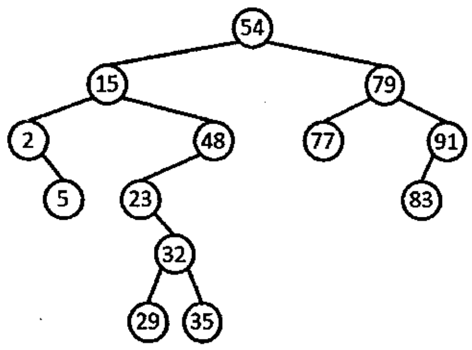


Fig.2 Splay tree

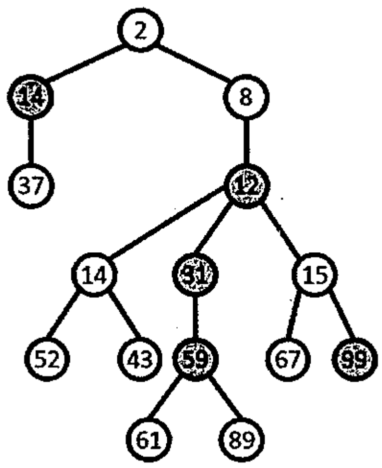


Fig.3 Fibonacci heap

試題隨卷繳回