國立政治大學112學年度碩士班暨碩士在職專班招生考試

電腦閱卷選擇題答案

系所組名稱：資訊科學系（資訊科學與工程組、智慧計算組）

科目名稱：資料結構及演算法

題號	答案	題號	答案
1	A		
2	D		
3	C		
4	D		
5	B		
6	D		
7	B		
8	A		
9	C		
10	A		
11	C		
12	D		

考試科目	資料結構及演算法	系所別	資訊科學系/資訊科學與工程組、智慧計算組	考試時間	2 月 3 日(五) 第 四 節

1. (60%) Select the best answer. 選擇題請在答案卡上作答，否則不予計分。

Unless otherwise stated, n is the number of data in a data structure.

1) (5%) Consider a linked list in which each node stores an integer.

Which of the following statements is true?

A. If the list is a doubly linked list, then inserting a node to the front of the list takes $O(1)$ time.

B. If the list is a singly linked list, then deleting a given node from the list takes $O(1)$ time.

C. If the list is a doubly linked sorted list (i.e., the ith node contains the ith smallest integer), then searching an integer in the list takes $O(\log n)$ time.

D. If the list is a singly-linked sorted list, then searching an integer in the list takes $O(\log n)$ time.

2) (5%) Let `arr` be an array that contains n (n>3) integers. Moreover, `arr[0]=arr[1]=arr[2]=n+1` and `arr[i]=i` for every $3 \leq i \leq n-1$.

What is the time complexity of sorting `arr` in non-decreasing order using insertion sort?

A. $\Theta(n^3)$.

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

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

D. $\Theta(n)$.

3) (5%) Which of the following statements is true?

A. The worst-case time complexity of counting sort is $\Theta(n \log n)$.

B. The worst-case time complexity of quick sort is $\Theta(n \log n)$.

C. The average-case time complexity of quick sort is $\Theta(n \log n)$.

D. The best-case time complexity of counting sort is $\Theta(n \log n)$.

4) (5%) Consider a variant of merge sort where merging two sorted arrays of sizes n_a and n_b $(n_a \geq n_b)$ takes $\Theta(n_a^2)$ time in the worst case. Let $T(n)$ be the worst-case time complexity of this variant of merge sort. Which of the following recurrence relations is true?

A. $T(n) = T(n/2) + \Theta(n)$.

B. $T(n) = 2T(n/2) + \Theta(n)$.

C. $T(n) = T(n/2) + \Theta(n^2)$.

D. $T(n) = 2T(n/2) + \Theta(n^2)$.

5) (5%) Consider again the above variant of merge sort. What is the worst-case time complexity?

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

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

C. $\Theta(n^2 \log n)$.

D. $\Theta(n^3)$.

考試科目	資料結構及演算法	系所別	資訊科學系/資訊科學與工程組、智慧計算組	考試時間	2月3日(五) 第四節

6) (5%) Which of the following statements about binary search trees is false?
   A. Tree height is always $\Omega(\log n)$.
   B. Given the root of the tree, there is an $O(n)$-time algorithm that sorts all the data in the tree.
   C. Given the root of the tree and an integer k in $\{1,2,\ldots,n\}$, there is an $O(hk)$-time algorithm that outputs the kth smallest data in the tree, where h is the tree height.
   D. The smallest data is always stored in a leaf node.

7) (5%) Which of the following statements about binary heaps is false?
   A. It is a binary tree.
   B. Given the root of the tree, there is an $O(n)$-time algorithm that sorts all the data in the tree.
   C. Tree height is always $\Theta(\log n)$.
   D. It can be implemented using an array.

8) (5%) Which of the following data structures is best suited for range query (i.e., output all the data in the data structure that are between x and y, where x and y are input numbers)?
   A. Red black tree.
   B. Stack.
   C. Fibonacci heap.
   D. Hash table.

9) (5%) Which of the following statements about hash tables is true?
   A. The best-case time complexity of inserting a key-value pair is $\Theta(\log n)$.
   B. After a hash table is created, the hash function cannot be changed.
   C. The best-case time complexity of searching a key is $O(1)$.
   D. Let h be the hash function. If $x \neq y$, then $h(x) \neq h(y)$.

10) (5%) Which of the following problem is NP-hard?
   A. Find the largest cycle in a graph.
   B. Find the largest component in a graph.
   C. Find the minimum spanning tree in an edge-weighted graph.
   D. Find the maximum spanning tree in an edge-weighted graph
   (i.e., find the spanning tree that has the largest total edge weight).

考試科目	資料結構及演算法	系所別	資訊科學系/資訊科學與工程組、智慧計算組	考試時間	2月3日(五)第四節

11) (5%) Which of the following statements is true?

A. If a problem is NP-hard, then no algorithm can solve the problem.

B. If a problem is NP-complete, then every algorithm needs $\Omega(2^n)$ time to solve the problem in the worst case, where n is the input size of the problem.

C. If there is a polynomial time algorithm that solves some NP-hard problem, then every NP problem has a polynomial time algorithm.

D. If a problem is NP-hard, then it is an NP problem.

12) (5%) Which of the following statements about graphs is false?

A. The sum of the degrees of all vertices is equal to twice the number of edges.

B. Let G be a connected graph, and let n be the number of vertices in G. Then G has at least $n-1$ edges.

C. If a graph is stored in an adjacency matrix, then it takes $O(1)$ time to check whether two given vertices are adjacent or not.

D. Let G be a graph stored in an adjacency matrix. Let d_{max} be the largest vertex degree in G. Then it takes $O(d_{max})$ time to compute the degree of any given vertex.

2. (20%) In the knapsack problem, the input contains n items $1,2,3,\ldots,n$, where item i has weight $w[i]$ and value $v[i]$. The input also contains a number W. All the weights, values, and W are positive integers. The output is a set $S \subseteq \{1,2,3,\ldots,n\}$ such that the total weight of S is at most W (i.e., $\sum_{i\in S} w[i] \leq W$) and the total value of S (i.e., $\sum_{i\in S} v[i]$) is maximized.

Consider the following subproblem:

$K(j,W')$: return the maximum achievable total value under the constraints that only items $1,2,3,\ldots,j$ can be chosen and the total weight of the chosen items is at most W'.

Thus, $K(n,W)$ returns the total value of the optimal solution of the original knapsack problem.

1) (10%) Design a dynamic programming algorithm (in pseudocode) for the knapsack problem based on the above subproblem. In your answer, please state the recursive formula for $K(j,W')$ clearly.

2) (5%) Please analyze the time complexity of your algorithm.

3) (5%) Is your algorithm a polynomial time algorithm? Please explain your answer.

3. (20%) Consider again the knapsack problem.

1) (10%) Design a greedy heuristic (in pseudocode) for the knapsack problem. Please explain why your heuristic can be considered to be greedy.

2) (5%) Please analyze the time complexity of your heuristic.

3) (5%) Does your heuristic always output the optimal solution? If so, please give a proof. If not, please give a toy example to show that your heuristic may not obtain the optimal solution.

備註	一、作答於試題上者，不予計分。 二、試題請隨卷繳交。