

本份考試卷共有三頁六大題
請於答案卷上標明題號後依序作答
請勿使用選擇題作答區

1. (30 points) *In this problem, and this problem only*, please choose among the following items and show your answers by the item labels.

[A] $O(1)$	[B] $O(n)$	[C] $O(\log n)$	[D] $O(n \log n)$
------------	------------	-----------------	-------------------

The vector data structure is basically a flexible-size array, i.e., the number of elements inside the vector can be unlimited. The vector data structure can be implemented with either the usual fixed-length array or a singly linked list.

The vector can be implemented by a fixed-length array with an auxiliary length information *without* wasting any other space of the array. Then, accessing an element within the vector by index is trivial. But in the worst case, when an element is inserted into the vector, a new array needs to be created and all the contents in the original array need to be moved; similarly, when an element is removed from the vector, the creating and moving steps need to be performed, too.

The vector can also be implemented by a singly linked list *without* wasting any other space. Then, inserting an element into or removing an element from the vector by index does not need moving any contents.

Assume there are n elements in a vector V , what is the asymptotic worst-case running time for each operation under different implementations?

	array	linked list
$Access(V, i)$	(1)	(2)
$Insert(V, x, i)$	(3)	(4)
$Delete(V, i)$	(5)	(6)

Here $Access(V, i)$ returns the element at the i -th position in V , $Insert(V, x, i)$ inserts an element x into the i -th position in V , and $Delete(V, i)$ deletes an element at the i -th position in V . Both $Insert$ and $Delete$ will change the size of V .

Besides the useful singly linked list, doubly linked list is a more continent data structure, since a node can have two links point to both its successor and predecessor. Consider a list L that consists of n objects with one root pointer and one tail pointer pointing to its first and last object. What is the asymptotic worst-case running time for each dynamic-set operation listed in the following table?

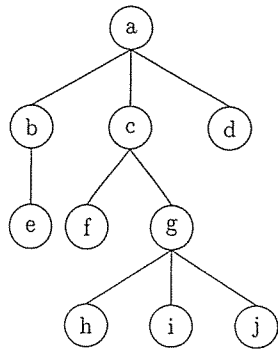
	singly linked list	doubly linked list
$Search(L, k)$	(7)	(8)
$Remove(L, p)$	(9)	(10)

Here $Search(L, k)$ returns the object whose key equals k in L , and $Remove(L, x)$ removes a node pointed by pointer p from L .

見背面

2. (20 points) Please answer the following short questions about trees. (Note: You do NOT need to prove the results.)

- (1) (2 points) What is the maximum number of nodes on level i of a binary tree?
- (2) (2 points) What is the maximum number of nodes in a binary tree of depth k ?
- (3) (2 points) For any non-empty binary tree T , Let n_0 be the number of leaf nodes and n_2 the number of nodes of degree 2. What the relationship between n_0 and n_2 ?
- (4) (2 points) What is the depth of a complete binary tree with n nodes?
- (5) (2 points) How many null links do we have in an n -node binary tree of linked representation?
- (6) (4 points) Draw the binary tree that corresponds to the left child-right sibling representation of the following tree.



- (7) (2 points) In order to access the tree nodes quickly, the binary tree in the previous question is stored level-by-level in a one-dimensional array $A[1..n]$. What are the indices of nodes g and j within the array?
 - (8) (2 points) Draw the binary tree if the preorder sequence is $[1, 2, 3, 4, 5, 6, 7]$ and the inorder sequence is $[2, 4, 3, 1, 6, 5, 7]$.
 - (9) (2 points) Draw the binary tree if the postorder sequence is $[7, 4, 2, 5, 6, 3, 1]$ and the inorder sequence is $[7, 4, 2, 1, 5, 3, 6]$.
3. (6 points) Please use a stack to illustrate how to evaluate the postfix expression "62/3-42*+" step by step. Please draw the stack status after each step.
4. (8 points) Please answer the following short questions about quick sort. (Note: Do NOT use the table in Question 1)
- (1) (3 points) What is the average time complexity of quick sort on n numbers?
 - (2) (3 points) What is the worst-case time complexity of quick sort on n numbers?
 - (3) (2 points) For using quick sort towards a non-decreasing sequence, if quick sort is implemented by always choosing the first component as the midpoint, for all permutations of $\{1, 2, \dots, 6\}$, what is the permutation that exhibits the worst case behavior of quick sort?

5. (18 points) (*Note: You do NOT need to prove the results.*) Prim's algorithm can be used for producing a minimum spanning tree. The essence of the algorithm involves greedily searching for a vertex that is connected by the lightest edge to but is not within the current tree. When running Prim's algorithm on a graph with V vertices and $E = O(V^{1.5})$ edges stored with the adjacency list representation, what is the worst case asymptotic time complexity of the algorithm with respect to V if
- (1) (6 points) no other data structures are used
 - (2) (6 points) a loser tree is used to store the vertices, where the leaves hold the smallest direct edge length to the current tree (or ∞ if a direct edge does not exist or the vertex is within the current tree)
 - (3) (6 points) a Fibonacci heap is used to store the vertices
6. (18 points) (*Note: You do NOT need to prove the results.*) The recursive closest-point finding algorithm is a classic divide and conquer algorithm for finding closest points on a Euclidean plane. The algorithm recursively separates the set of points to two equal-sized subsets according to their x -axis, and find the closest points within the subsets as well as between the subsets. For a set of N points, what is the asymptotic time complexity of the algorithm with respect to N if
- (1) (6 points) finding the closest points between two size- $\frac{N}{2}$ subsets take $O(N)$ time
 - (2) (6 points) finding the closest points between two size- $\frac{N}{2}$ subsets take $O(N \log N)$ time
 - (3) (6 points) finding the closest points between two size- $\frac{N}{2}$ subsets take $O(N^2)$ time

試題隨卷繳回