

考試科目	資料結構及演算法	系所別	資訊科學系	考試時間	
<p>1 (20 points) What are the worst case time complexities of the following algorithms?</p> <ol style="list-style-type: none"> <li>1.1 Binary search in a sorted array consisting of <math>n</math> integers</li> <li>1.2 Building a binary heap from an unsorted array consisting of <math>n</math> integers</li> <li>1.3 Randomized quicksort for an array consisting of <math>n</math> integers</li> <li>1.4 Kruskal algorithm with disjoint-set-forest for finding the minimum spanning tree given a connected, undirected graph with <math> V </math> vertices and <math> E </math> edges</li> </ol> <p>2 (20 points) Design a dynamic array that supports the following four operations in <math>\Theta(1)</math> amortized time. Show your algorithm for each of the four operations in C or pseudo code.</p> <ol style="list-style-type: none"> <li>2.1 Initializing an empty array</li> <li>2.2 Inserting a new element at the end of the array</li> <li>2.3 Removing the last element of the array</li> <li>2.4 Access to the <math>k</math>th element of the array</li> </ol> <p>3 (20 points) Given two binary trees, design an algorithm for checking if they are identical in terms of both the structure and the values. The node in the binary tree is defined as follows. Implement the "equal" function in C or pseudo code.</p> <pre> struct node {     int value;     struct node *left, *right; }; bool equal(struct node *tree1, struct node *tree2) {     .... } </pre> <p>4 (20 points) Given an <math>m \times n</math> matrix <math>M</math> of real numbers, find a <i>maximum sum submatrix</i> of <math>M</math>. The maximum sum submatrix of <math>M</math> is a submatrix of <math>M</math> with the largest sum. Describe your algorithm in C or pseudo code and show its time complexity.</p> <p>5 (20 points) Given a string <math>s</math> consisting of <math>n</math> characters and a dictionary <math>D</math> consisting of <math>m</math> words, provide an algorithm for checking if <math>s</math> is completely composed of the words in <math>D</math>. That is, <math>s</math> can be segmented into <math>k</math> fragments, and all the <math>k</math> fragments are listed in <math>D</math>. Describe your algorithm in C or pseudo code and show its time complexity.</p>					
備註	<p>一、作答於試題上者，不予計分。 二、試題請隨卷繳交。</p>				