傳播管理

銘傳大學九十一學年度資訊管理研究所碩士班招生考試

第二節

資料結構 試題

1. (12%) For each tree given below, choose all the descriptions that apply from the list below:

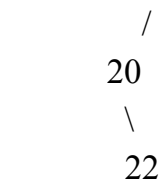
   i binary tree

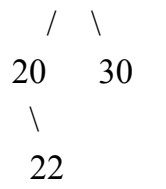
   ii binary search tree

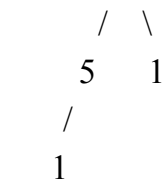
   iii. AVL tree

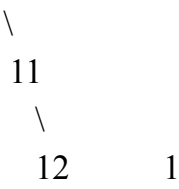
   iv. complete binary tree

   v. heap

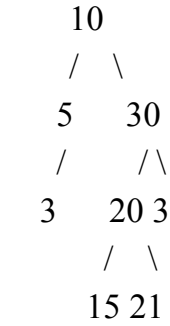
   (A)
   

   (B)
   

   (C)
   

   (D)
   

2. (5%) Show what the following AVL tree looks like after inserting 12.



3. (15%)

   (A) (5%) Briefly explain why sorting with comparisons requires $\Omega(n \log n)$ time.

   (B) (5%) Name a linear time sort algorithm.

   (C) (5%) Determine the big-O operation count for the execution of the following

   code fragment (in terms of N). An exact operation count is not required.

```
int Sum = 0;
for (int I = l; i <=N; i = 2*i){
for (int j = l; j <= N; j++){
Sum = Sum + i*j;
}
```

}

4. (5%) Rank the following functions in ascending order (slowest-growth first):

$$\lg(n),\ n,\ n^2,\ 2^n,\ n!,\ n^3,\ n.\ \lg(n),\ n^{1/2}$$

5. (8%) Match each hashing method below with its main disadvantage(s). One or more disadvantages may apply.

   Disadvantages:

   (1) clustering results if many keys hash to the same value
   (2) clustering results if many keys hash to similar but different values
   (3) requires extra memory beyond the hash table itself
   (4) most expensive to compute the hash function
   (5) can fail to find a free spot for an insert even if the table is not full and the table size is prime

   Hashing methods (enter the number(s) of the disadvantage)

   (A) separate chaining
   (B) linear Probing
   (C) quadratic probing
   (D) double hashing

6. (10%) Match each sorting method with its main advantage(s). One or more advantage may apply

   Advantages:

   (1) best for external sorting
   (2) fastest on input that is already nearly sorted
   (3) best asymptotic complexity for fixed-length keys
   (4) in practice fastest general sorting method for main-memory sorts
   (5) optimal asymptotic complexity among comparison-based methods

   Sorting method (enter the number(s) of the advantage)

   (A) insertion sort
   (B) heap sort
   (C) merge sort
   (D) quick sort
   (E) radix sort

7. (20%) Consider how the graph abstract data type might be used to represent friendships among groups of people. Each vertex could correspond to a person, and an edge beaten two vertices could represent a friendship.

   Now suppose that the following friendships exist:

   Alan is friends with Bob, Chuck and Dave

Chuck and Dave are friends
Ed is friends with Bob and Fred
Gary and Hal are friends
Irving is friends With Gary

(A) (4%) Draw the graph that represents exactly the friendships listed above.

(B) (3%) How many:
i. vertices
ii edges
iii components
are there in the graph you drew for part 7A?

(C) (3%) Suppose that all the people mentioned in the list of friendships were friends with one another. How many friendships would there be? Briefly explain your answer.

(D) (5%) On this graph from part 7A, show the labeling of vertices and edges that would result from performing breadthFirstSearch(Alan).

(E) (5%) On this graph from part 7A, show the labeling of vertices and edges that would result from performing depthFirstSearch(Alan).

8. (20%) Answer the following TRUE/ALSE questions: (+2 per correct answer, 0 per missing, -3 per incorrect)

(1) When using “binary search” to search an array, the array must always be sorted in increasing order.

(2) When using "binary search" to search an array, and the arrays is not sorted, then the program will crash.

(3) When using "binary search" to search an mp, and the arrays is not sorted, then you will never find the value you are looking for even if it does exist in the array.

(4) When using "binary search" to search an array, and the arrays is not sorted, then sometimes the search will be successful, but sometimes it will fail even if the value does exist in the array.

(5) A tree is an example of a list.

(6) A list is an example of a tree.

(7) Suppose you insert the values 40, 30, 60, 50, and 10 into a binary search tree, and then use the following function:

```
void someOrder(Node*p)
{ if (p ! = 0)
     { someOrder(p->right);
          cout << p->data << endl; someOrder(P->left);
```

```
        }
    }
```

then the output of calling someOrder(root) would be: 10, 30, 40, 50, then 60.

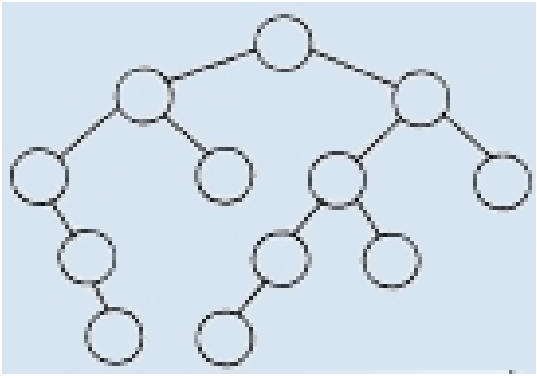
(8) A heap is implemented as a two-dimensional array.

(9) $5n^2 + 3n + 2$ is $\Omega((\log n)^2)$

(10) The best-case runtime of removal from heap-based priority queue is O(l).

9. (5%) Insert keys into the binary tree structure given so that a postorder traversal will give:

8	3	5	11	14	4	2	1	9	7	6	0



試題完