ADSAA MOOC 2021 - Midterm Exam

开始时间 2022/04/15 17:00:00　　结束时间 2022/04/29 12:00:00　　答题时长 45分钟

答卷类型 标准答案　　总分 100

判断题

得分：暂无　总分：30

1-1 When measuring the relevancy of the answer set, if the precision is low but the recall is high, it means that most of the relevant documents are missing, but most of the retrieved documents are relevant. (3分)

○ T ◉ F

1-2 In amortized analysis, a good potential function should always assume its minimum at the start of the sequence. (3分)

◉ T ○ F

1-3 In an AVL tree, it is possible to have this situation that the balance factors of a node and both of its children are all -1. (3分)

◉ T ○ F

1-4 The root of a B+ tree of order m has at most m subtrees. (3分)

◉ T ○ F

1-5 Finding the maximum key from a splay tree will result in a tree with its root having no left subtree. (3分)

○ T ◉ F

1-6 In a red-black tree, an internal red node cannot be a node of degree 1. (3分)

◉ T ○ F

1-7 To solve a problem by dynamic programming instead of recursions, the key approach is to store the results of computations for the subproblems so that we only have to compute each different subproblem once. Those solutions can be stored in an array or a hash table. (3分)

◉ T ○ F

1-8 Making N insertions into an initally empty binomial queue takes $\Theta(N \log N)$ time in the worst case. (3分)

○ T ◉ F

1-9 The Huffman code is one kind of optimal prefix codes. For a given alphabet and its characters' frequencies, the Huffman codes may not be unique, but the Huffman code **length** of each character is unique. (3分)

○ T ◉ F

1-10 For the recurrence equation $T(N) = aT(N/b) + f(N)$, if $af(N/b) = f(N)$, then $T(N) = \Theta(f(N) log_b N)$. (3分)

◉ T ○ F

单选题

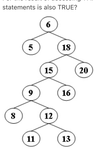
得分：暂无　总分：50

2-1 Given 4 cases of frequences of four characters. In which case(s) that the total bits taken by Huffman codes are the same as that of the ordinary equal length codes? (5分)

- (1) 1 2 2 3
- (2) 1 1 1 2
- (3) 2 2 3 5
- (4) 1 2 3 4

◉ A. (1) and (2)
○ B. (3) only
○ C. (1), (2) and (4)
○ D. (2) only

2-2 For the result of accessing 11 in the splay tree in the following figure, besides saying that 11 must be the root, which one of the following statements is also TRUE? (5分)

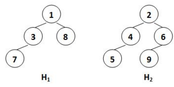


◉ A. 12 and 18 are siblings
○ B. 9 and 18 are siblings
○ C. 6 and 18 are siblings
○ D. 12 is a leaf node

2-3 When solving a problem with input size N by divide and conquer, if at each step, the problem is divided into 4 sub-problems and each size of these sub-problems is $N/2$, and they are conquered in $O(N^2 log N)$. Which one of the following is the closest to the overall time complexity? (5分)

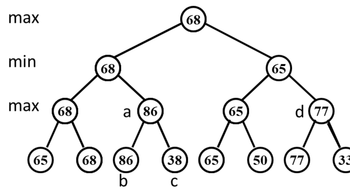
○ A. $O(N^2 log N)$
○ B. $O(N^2)$
○ C. $O(N^2 log N)$
◉ D. $O(N^2 log^2 N)$

2-4 Merge the two skew heaps in the following figure. Which one of the following statements is FALSE? (5分)



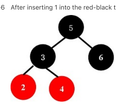
○ A. the null path length of 6 is the same as that of 2
○ B. 1 is the root with 3 being its right child
○ C. Along the left most path from top down, we have 1, 2, 6, and 8
◉ D. 5 is the right child of 4

2-5 Given the following game tree, which node is the first one to be pruned with α-β pruning algorithm? (5分)



○ A. a
○ B. b
◉ C. c
○ D. d

2-6 After inserting 1 into the red-black tree given in the figure, which node(s) will keep its/their color(s) unchanged? (3分)



○ A. both 3 and 6
◉ B. both 5 and 6
○ C. 3, 5, and 6
○ D. 3, 4, 5, and 6

2-7 There are 8000 documents in the database. The statistic data for one query are shown in the following table. The precision is: __ (5分)

	Relevant	Irrelevant
Retrieved	1000	1000
Not Retrieved	2000	4000

○ A. 12.5%
○ B. 20%
○ C. 33%
◉ D. 50%

2-8 To solve the optimal binary search tree problem, we have the recursive equation $c_{ij} = \min_{i \le k \le j} \{w_{ij} + c_{i,j-1} + c_{i+1,j}\}$. To solve this equation in an iterative way, we must fill up a table as follows: (5分)

○ A.
```
for i= 1 to n-1 do;
  for j= i to n do;
    for l= i to j do
```

○ B.
```
for j= 1 to n-1 do;
  for i= 1 to j do;
    for l= i to j do
```

◉ C.
```
for k= 1 to n-1 do;
  for i= 1 to n-k do;
    set j = i+k;
    for l= i to j do
```

○ D.
```
for k= 1 to n-1 do;
  for i= 1 to n do;
    set j = i+k;
    for l= i to j do
```

2-9 When doing amortized analysis, which one of the following statements is FALSE? (2分)

○ A. Aggregate analysis shows that for all n, a sequence of n operations takes worst-case time $T(n)$ in total. Then the amortized cost per operation is therefore $T(n)/n$

○ B. For accounting method, when an operation's amortized cost exceeds its actual cost, we save the difference as credit to pay for later operations whose amortized cost is less than their actual cost

◉ C. For potential method, a good potential function should always assume its maximum at the start of the sequence

○ D. The difference between aggregate analysis and accounting method is that the later one assumes that the amortized costs of the operations may differ from each other

2-10 Insert key 48 into the balanced binary tree shown by the figure. Then in the resulting balanced tree, the left- and right-child of key 37 are: (5分)

○ A. 13 and 48
○ B. 24 and 48
◉ C. 24 and 53
○ D. 24 and 90

2-11 Insert { 5, 1, 7, 8, 21, 2, 12, 19, 13, 0 } into an initially empty 2-3 tree (with splitting). Which one of the following statements is FALSE? (5分)

◉ A. 13 and 19 are in the same node
○ B. the parent of the node containing 8 has 3 children
○ C. the first key stored in the root is 12
○ D. there are 5 leaf nodes

程序填空题

得分：暂无　总分：20

5-1 The function `DeleteRt` is to delete the root of a subtree with index `Pos` from a binomial queue `H`. The rest of the subtree is then stored as a new binomial queue and returned.

```
BinQ DeleteRt( BinQ H, int Pos )
{
    BinTree OldRoot, SubTree;
    BinQ NewBinQ;
    int p;

    OldRoot = H->TheTrees[Pos];
    SubTree = OldRoot->LeftChild;
    free(OldRoot);
    NewBinQ = Initialize();
    NewBinQ->CurrentSize = (1<<Pos) - 1 (5分);
    for ( p=Pos-1; p>=0; p-- (5分) ) {
        NewBinQ->TheTrees[p] = SubTree;
        SubTree = SubTree->NextSibling;
        NewBinQ->TheTrees[p]->NextSibling = NULL;
    }
    return NewBinQ;
}
```

5-2 A binary tree is said to be "height balanced" if both its left and right subtrees are height balanced, and the heights of its left and right subtrees can differ by at most 1. That is, $|H_L - H_R| \le 1$ where H_L and H_R are the heights of the left and right subtrees, respectively. An empty binary tree is defined to be height balanced.

The function `IsBalanced` is to judge if a given binary tree `T` is height balanced. If the answer is yes then return `true` and store the tree height in the parameter `pHeight`, else simply return `false`. The height of an empty tree is defined to be 0.

```
typedef struct TNode *BinTree;
struct TNode{
    int Key;
    BinTree Left;
    BinTree Right;
};

bool IsBalanced ( BinTree T, int *pHeight )
{
    int  LHeight, RHeight, diff;

    if( T == NULL) {
        *pHeight = 0;
        return true;
    }
    else if ( IsBalanced(T->Left, &LHeight) && IsBalanced(T->Right, &RHeight) ) {
        diff = LHeight - RHeight;
        if ( diff<=1 && diff>= -1 (5分) )  {
            *pHeight = 1 + ( diff<0 ? RHeight : LHeight (5分) );
            return true;
        }
        else return false;
    }
    return false;
}
```