历年卷1

开始时间	2023/06/20 10:18:00	**结束时间**	2023/06/29 20:59:00	**答题时长**	13601分钟
答卷类型	标准答案	**总分**	100		

判断题

得分：暂无 总分：26

1-1 The n-th Fibonacci number can be computed by divide and conquer method of computing x^n, where x is the matrix (2分)

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Then the n^2-th Fibonacci number F_n2 can be computed in $O(\log n)$ time.

● T ○ F

1-2 If we build a skip list for n elements, then we can search or delete any element in $O(\log n)$ time in the worst case.(2分)

○ T ● F

1-3 If an operation takes $O(1)$ expected time, then it takes $O(1)$ amortized time.(2分)

○ T ● F

1-4 The number of light nodes along the right path of a skew heap is $O(\log N)$.(2分)

● T ○ F

1-5 Debugging parallel programs is simpler than normal sequential programs.(2分)

○ T ● F

1-6 What makes the time complexity analysis of a backtracking algorithm very difficult is that the sizes of solution spaces may vary. (2分)

○ T ● F

1-7 A binary tree that is not full cannot correspond to an optimal prefix code.(2分)

● T ○ F

1-8 NP-hard problems and NP-complete problems are the subsets of NP problems.(2分)

○ T ● F

1-9 A word which has high term frequency in the whole document set must be a stop word.(2分)

○ T ● F

1-10 When insert three keys into a non-empty 2-3 tree, and if the tree gains height when the first key is in, then we can be sure that the 2-3 tree will **not** gain height after the insertions of the next two keys. (2分)

● T ○ F

1-11 Suppose that the replacement selection is applied to generate longer runs for N numbers with a priority queue of size M, the possible maximum length of the longest run is N. (2分)

● T ○ F

1-12 For an optimization problem, given a neighborhood, if its local optimum is also a global optimum, one can reach an optimal solution with just one step of local improvements. (2分)

○ T ● F

1-13 Revisit the activity selection problem. Given a set of activities $S = \{a_1, a_2, \cdots, a_n\}$ that wish to use a resource, (2分) each a_i takes place during a time interval. The goal is to arrange as many compatible activities as possible. Recall that several greedy approaches are introduced in the class, among which the one selecting an activity with the shortest length, denoted by SF, is not always optimal. However, we claim that SF accepts at least $|OPT|/2$

activities, given that the optimal value is $|OPT|$, where OPT is an optimal solution. Check if the following is a correct proof.

We use a technique, called the charging scheme, similarly as the amortized analysis. Suppose each accepted activity of OPT holds one dollar, which will be given to the activities accepted by SF in the following way. For any activity a_i of OPT, if a_i is also accepted by SF, give the dolloar to itself. Otherwise, there must be some activity a_j, accepted by SF, is not compatible with a_i. Then a_j receives one dollar from a_i. Along this line, each activity of OPT sends out one dollar to an activity in SF, while each activity of SF receives at most two dollars. It implies that SF accepts as least $|OPT|/2$ activities.

● T ○ F

单选题

得分：暂无 总分：60

2-1 We have m balls and m boxes. Each ball is assigned to one of the m boxes independently and uniformly at random (i.e., equally likely to each box). Suppose that m is sufficiently large, and e is the natural number, which of the following is FALSE? (3分)

○ A. The expected number of balls in a box is 1.

○ B. The epected number of empty boxes is m/e.

○

C. Suppose that one box can only contain one ball, and it will reject other balls if it already contains one. Then, the expected number of rejected balls is m/e.

●

D. Suppose a box can accommodate two balls, and will reject any additional balls. The expected number of boxes containing exactly two balls is m/e.

2-2 Which of the asymptotic upper bound for the following recursive $T(n)$ is correct? (3分)

○ A. $T(n) = 2T(n/2) + n\log^2 n$. Then $T(n) = O(n\log^2 n)$.

● B. $T(n) = T(n^{1/3}) + T(n^{2/3}) + \log n$. Then $T(n) = O(\log n \log\log n)$

○ C. $T(n) = 3T(n/2) + n$. Then $T(n) = O(n)$.

○ D. $T(n) = 2T(\sqrt{n}) + \log n$. Then $T(n) = O(\log n)$.

2-3 Consider two disjoint sorted arrays $A[1\ldots m]$ and $B[1\ldots n]$, we would like to compute the k-th smallest element in the union of the two arrays, where $k \le \min\{m, n\}$. Please choose the smallest possible running time among the following options. (3分)

● A. $O(\log k)$

○ B. $O(\log m)$

○ C. $O(\log n)$

○ D. $O(\log m + \log n)$

2-4 In this problem, we would like to find the amortized cost of insertion in a dynamic table T. Initially the size of the table T is 1. The cost of insertion is 1 if the table is not full. When an item is inserted into a full table, the table T is expanded as a new table of size 3 times larger. Then, we copy all the elements of the old table into this new table, and insert the item in the new table. (3分)

Let $num(T)$ be the number of elements in the table T, and $size(T)$ be the total number of slots of the table. Clearly, if the table T is full, the cost of one insertion is $num(T) + 1$.

Let D_i denote the table after applying the ith operation on D_{i-1}.
Which of the following potential function $\Phi(D_i)$ can help us achieve $O(1)$ amortized cost per insertion?

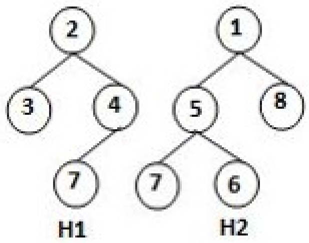
○ A. $\Phi(D_i) = num(T) - \frac{size(T)}{3}$

● B. $\Phi(D_i) = \frac{3}{2}(num(T) - \frac{size(T)}{3})$

○ C. $\Phi(D_i) = \frac{3}{2}(num(T) + \frac{size(T)}{3})$

○ D. $\Phi(D_i) = num(T) + \frac{size(T)}{3}$

2-5 Merge the two leftist heaps in the following figure. Which one of the following statements is FALSE? (3分)



○ A. 4 and 3 are siblings

● B. along the left path from the root, we have 1, 2, 4, and 8

○ C. 6 is the right child of 5

○ D. 2 and 4 have the same NPL

2-6 In typical applications of data structures, it is not a single operation that is performed, but rather a sequence of operations, and the relevant complexity measure is not the time taken by one operation but the total time of a sequence. Hence instead of imposing any explicit structural constraint, we allow the data structure to be in an arbitrary state, and we design the access and update algorithms to adjust the structure in a simple, uniform way, so that the efficiency of future operations is improved. We call such a data structure **self-adjusting**. For example skew heaps and splay trees are such kind of structures. (3分)

Which one of the following statements is FALSE about self-adjusting data structures?

○ A. They need less space, since no balance information is kept.

○ B. Their access and update algorithms are easy to understand and to implement.

○ C. In an amortized sense, ignoring constant factors, they can be at least as efficient as balanced structures.

● D. Less local adjustments take place than in the corresponding balanced structures, especially during accesses.

2-7 The Merging problem is to merge two non-decreasing arrays A(1), A(2), ..., A(n) and B(1), B(2), ..., B(n) into another non-decreasing array C(1), C(2), ..., C($2n$). To solve it in parallel, we turn it into a Ranking problem. That is, to compute RANK(A(i),B) and RANK(B(i),A) for every $1 \le i \le n$, where RANK(e,S) is the position of e in S. The following psuedo-code is for solving the Ranking problem parallely. (3分)

```
for Pi, 1<=i<=n  pardo
    RANK(A(i),B) := BS(A(i),B)
    RANK(B(i),A) := BS(B(i),A)
```

where `BS(e,S)` is to find the position of `e` in `S` by **binary search**.

Which of the following gives the time and work load of the algorithm?

● A. $T(n) = O(\log n)$, $W(n) = O(n \log n)$

○ B. $T(n) = W(n) = O(n \log n)$

○ C. $T(n) = W(n) = O(n)$

○ D. $T(n) = O(\log n)$, $W(n) = O(n)$

2-8 Consider the problem of making change for n cents using the fewest number of coins. Assume that each coin's value is an integer. (3分)
The coins of the lowest denomination（面额） is the cent.

(I) Suppose that the available coins are quarters (25 cents), dimes (10 cents), nickels (5 cents), and pennies (1 cent). The greedy algorithm always yields an optimal solution.

(II) Suppose that the available coins are in the denominations that are powers of c, that is, the denominations are c^0, c^1, ..., c^k for some integers $c > 1$ and $k >= 1$. The greedy algorithm always yields an optimal solution.

(III) Given any set of k different coin denominations which includes a penny (1 cent) so that there is a solution for every value of n, greedy algorithm always yields an optimal solution.

Which of the following is correct?

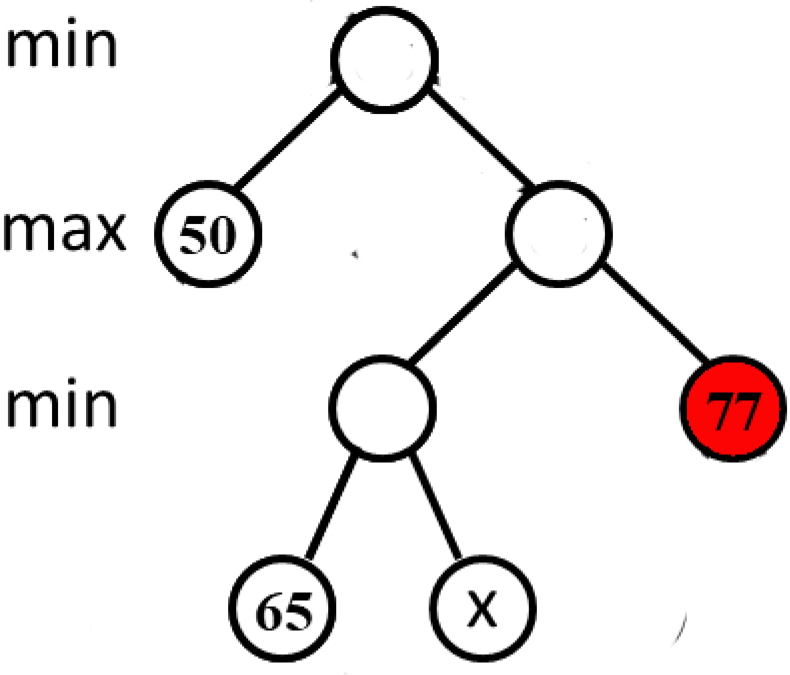
○ A. Statement (I) is false.

○ B. Statement (II) is false.

● C. Statement (III) is false.

○ D. All of the three statements are correct.

2-9 Given the following game tree, the red node will be pruned with α-β pruning algorithm if and only if ___. (3分)



● A. $x \geq 50$
○ B. $x \leq 65$
○ C. $x \geq 77$
○ D. $65 \leq x \leq 77$

2-10 About Vertex Cover problem, which of the following statements is FALSE? (3分)

○ A. It is an NP problem.
○ B. The time complexity of its verification algorithm is $O(N^3)$, where N refers to the number of nodes.
● C. It is polynomial-time reducible to Clique problem, but not vice versa.
○ D. It is an NP-complete problem.

2-11 Which of the following statements is FALSE? (3分)

○ A. All the P problems are NP problems.
● B. There exists an NP-complete problem such that not all the NP problems can be polynomially reduced to it.
○

C. For a problem, if we can find an algorithm which can solve it in polynomial time, then the problem is a P problem.

○ D. Not all decidable problems are in NP.

2-12 Given a set of 10000 emails in the mailbox, our task is to retrieve the spam emails from this set. The statistic data for a spam filter's performance are shown in the following table. The recall of this filter is: ___. (3分)

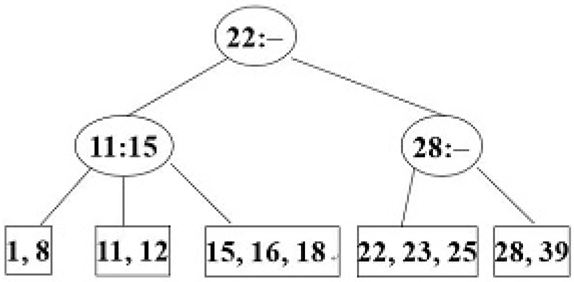
Category	**Actual True Spam**	**Actual False Spam**
Retrieved True Spam	2000	1000
Retrieved False Spam	2000	5000

○ A. 33.3%
○ B. 66.7%
● C. 50.0%
○ D. 28.6%

2-13 In a binomial queue, the total number of the nodes at even depth is always ___ than that of the nodes at odd depth (the root is defined to be at the depth 0). (3分)

● A. not smaller
○ B. not larger
○ C. smaller
○ D. larger

2-14 Given a 2-3 tree as shown in the following figure. Which of the following pairs of insertions will result in a 2-3 tree with different structures? (3分)



○ A. Inserting 9 vs. Inserting 13

○ B. Inserting 13 vs. inserting 30

○ C. Inserting 17 vs. inserting 19

● D. Inserting 19 vs. inserting 24

2-15 Two red-black trees are said to be different if they have different tree structures or different node colors. How many different red-black trees are there with 3 internal nodes? (3分)

○ A. 1 ● B. 2

○ C. 3 ○ D. more than 3

2-16 To sort N numbers by external sorting using a k-way merge and a k-size heap, which statement is TRUE about the total comparison times $T(N,k)$ and k? (3分)

● A. $T(N,k)$ has nothing to do with k.

○ B. $T(N,k)$ is $O(k)$ for fixed N.

○ C. $T(N,k)$ is $O(k\log k)$ for fixed N.

○ D. $T(N,k)$ is $O(k^2)$ for fixed N.

2-17 Among the following 6 statements about AVL trees and splay trees, how many of them are correct? (3分)

(1) AVL tree is a kind of height balanced tree. In a legal AVL tree, each node's balance factor can only be 0 or 1.

(2) Define a single-node tree's height to be 1. For an AVL tree of height h, there are at most $2^h - 1$ nodes.

(3) Since AVL tree is strictly balanced, if the balance factor of any node changes, this node must be rotated.

(4) In a splay tree, if we only have to find a node without any more operation, it is acceptable that we don't push it to root and hence reduce the operation cost. Otherwise, we must push this node to the root position.

(5) In a splay tree, for any non-root node X, its parent P and grandparent G (guranteed to exist), the correct operation to splay X to G is to rotate X upward twice.

(6) Splaying roughly halves the depth of most nodes on the access path.

● A. 2 ○ B. 3 ○ C. 4 ○ D. 5

2-18 Consider the **Minimum Degree Spanning Tree** problem: Given a connected undirected graph $G(V,E)$, find a spanning tree T whose maximum degree over its vertices is minimized over all spanning trees of G. The problem can be shown to be NP-hard by reduction from the Hamiltonian Path Problem. On the other hand, we can use local search to design approximating algorithms. Denote $d(u)$ as the degree of vertex u on a tree T. Consider the following algorithm: (3分)

1. Find an arbitrary spanning tree T of G.
2. If there's some edge $e \in E(G)\backslash E(T)$ with endpoints u, v, and there's some other vertex w on the path between u, v on T such that $max\{d(u), d(v)\} + 1 < d(w)$, then we replace an edge e' incident to w on T with e, i.e. $T := T \cup \{e\} \backslash \{e'\}$.
3. Repeat Step (2) until there's no edge to replace.

It can be shown that this algorithm will terminate at a solution with maximum vertex degree $OPT + O(\log |V|)$. To show the algorithm will terminate in finite steps, a useful technique is to define a nonnegative potential function $\phi(T)$ and to show $\phi(T)$ is strictly decreasing after each step. Which of the following potential functions below satisfies the above requirements?

○ A. $\phi(T) = \sum_{v\in V} d(v)$.

○ B. $\phi(T) = \sum_{(u,v)\in E(T)} \max\{d(u), d(v)\}$.

○ C. $\phi(T) = \sum_{u\in V} \sum_{v\in V, v\neq u} \sum_{w\in V, w\neq u,v} \max\{d(u), d(v), d(w)\}$.

● D. $\phi(T) = \sum_{v\in V} 3^{d(v)}$.

2-19 Consider the bin packing problem which uses a minimum number of bins to accommodate a given list of items. Recall that Next Fit (NF) and First Fit (FF) are two simple approaches, whose (asymptotic) approximation ratios are 2 and 1.7, respectively. Now we focus on a special class $I2$ of instances in which only two distinct item sizes appear. Check which of the following statements is true by applying NF and FF on $I2$. (3分)

○ A. NF and FF both have improved approximation ratios.

○ B. NF has an improved approximation ratio, while FF does not.

● C. FF has an improved approximation ratio, while NF does not.

○ D. Neither of NF or FF has an improved approximation ratio.

2-20 If there are 14 nodes in an AVL tree, then the maximum depth of the tree is ____. The depth of an empty tree is defined to be 0. (3分)

○ A. 3

○ B. 4

● C. 5

○ D. 6

程序填空题 得分：暂无 总分：6

5-1 The function `BinQueue_Merge` is to merge two binomial queues `H1` and `H2`, and return `H1` as the resulting queue.

```
BinQueue BinQueue_Merge( BinQueue H1, BinQueue H2 ) {
    BinTree T1, T2, Carry = NULL;
    int i, j;
    H1->CurrentSize += H2-> CurrentSize;
    for ( i=0, j=1; j<= H1->CurrentSize; i++, j*=2 ) {
        T1 = H1->TheTrees[i]; T2 = H2->TheTrees[i];
        switch( 4*!!T1 + 2*!!T2 + !!Carry (3分) ) {
        case 0:
        case 4: break;
        case 3:
        case 7:
                Carry = CombineTrees( T2, Carry ) (3分);
                H2->TheTrees[i] = NULL; break;
        case 1:
                H1->TheTrees[i] = Carry; Carry = NULL; break;
        case 2:
                H1->TheTrees[i] = T2; H2->TheTrees[i] = NULL; break;
        case 5:
                Carry = CombineTrees( T1, Carry );
                H1->TheTrees[i] = NULL; break;
        case 6:
                Carry = CombineTrees( T1, T2 );
                H1->TheTrees[i] = H2->TheTrees[i] = NULL; break;
        } /* end switch */
    } /* end for-loop */
    return H1;
}
```

编程题 得分：暂无 总分：8

7-1 Hand-made Cream (8分)

Suppose you are a baker planning to bake some hand-made cream breads.

To bake a cream bread, we need to use one slice of bread and one kind of cream. Each hand-made cream bread has a taste score to describe how delicious it is, which is obtained by multiplying the taste score for bread and the taste score for cream. (The taste scores could be negative, howerver, two negative tast scores can still produce delicious cream bread.)

The bread and cream are stored separately.

The constraint is that you need to examine the breads in a given order, and for each piece of bread, you have to decide immediately (without looking at the remaining breads) whether you would like to take it.

After you are finished with breads, you will take the same amount of cream in the same manner. The breads and creams you take will be paired in the same order as you take them.

Given N taste scores for bread and M taste scores for cream, you are supposed to calculate the maximum total taste scores for all hand-made cream bread.

Input Specification:

Each input file contains one test case. For each case, the first line contains two integers N and M ($1 \le N, M \le 10^3$), which are the numbers of bread and cream, respectively.

The second line gives N taste scores for bread.

The third line gives M taste scores for cream.

The taste scores are integers in $[-10^3, 10^3]$.

All the numbers in a line are separated by a space.

Output Specification:

Print in a line the maximum total taste score.

Sample Input:

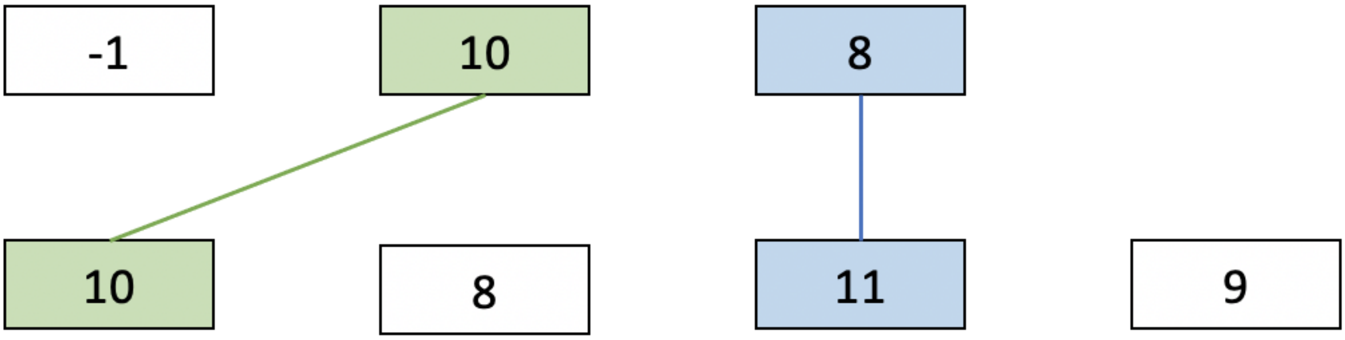
```
3 4
-1 10 8
10 8 11 9
```

Sample Output:

```
188
```

Hint:

The maximum total taste score for the sample case is $10 \times 10 + 8 \times 11 = 188$.


-1
10
8
10
8
11
9