▶ 1-1 分数 5 作者 陈越 单位 浙江大学

If a problem can be solved by dynamic programming, it must be solved in polynomial time.

○ T ◉ F

评测结果 答案正确

得分 5 分

1-2 分数 5 作者 陈越 单位 浙江大学

In a B+ tree, leaves and nonleaf nodes have some key values in common.

◉ T ○ F

评测结果 答案正确

得分 5 分

1-3 分数 5 作者 陈越 单位 浙江大学

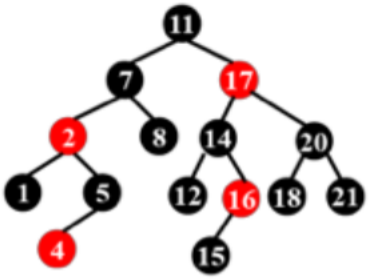
With the same operations, the resulting leftist heap is always more balanced than the skew heap.

○ T ◉ F

评测结果 答案正确

1-4 分数 5 作者 杨洋 单位 浙江大学

The following binary search tree is a valid red-black tree.



T F

评测结果 答案错误

得分 0 分

1-5 分数 5 作者 陈越 单位 浙江大学

What makes the time complexity analysis of a backtracking algorithm very difficult is that the sizes of solution spaces may vary.

T F

评测结果 答案错误

得分 0 分

1-6 分数 5 作者 张国川 单位 浙江大学

Recall the amortized analysis for Splay Tree and Leftist Heap, from which we can conclude that the amortized cost (time) is never less than the average cost (time).

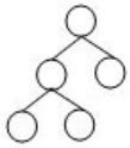
○ T ◉ F

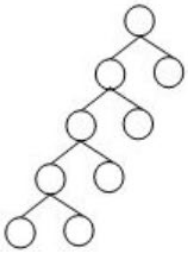
评测结果 答案正确

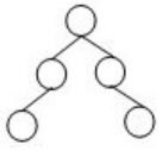
得分 5 分

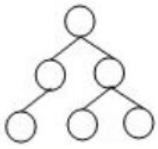
2-1 分数 5 作者 陈越 单位 浙江大学

Among the following trees, which one is impossible to be colored into a legal red-black tree?

○ A. 

◉ B. 

○ C. 

○ D. 

评测结果 答案正确

得分 5 分

2-2 分数 5 作者 陈越 单位 浙江大学

Which of the following is NOT a step in the process of building an inverted file index?

- A. Read in strings and parse to get words
- B. Use stemming and stop words filter to obtain terms
- C. Check dictionary with each term: if it is not in, insert it into the dictionary
- D. Get the posting list for each term and calculate the precision

评测结果 答案正确

得分 5 分

2-3 分数 5 作者 DS课程组 单位 浙江大学

For a skew heap, _____ is correct.

- A. Skew heap is a kind of leftist heap.
- B. The right path of a skew heap can be arbitrarily long.
- C. When running a sequence of merging operations, comparing to leftist heaps, skew heaps are always more efficient in running time for every merge.
- D. Comparing to leftist heaps, skew heaps are not always more efficient in space.

评测结果 答案正确

得分 5 分

2-4 分数 5 作者 DS课程组 单位 浙江大学

A binomial queue of size 42 can be represented by the following binomial queues __.

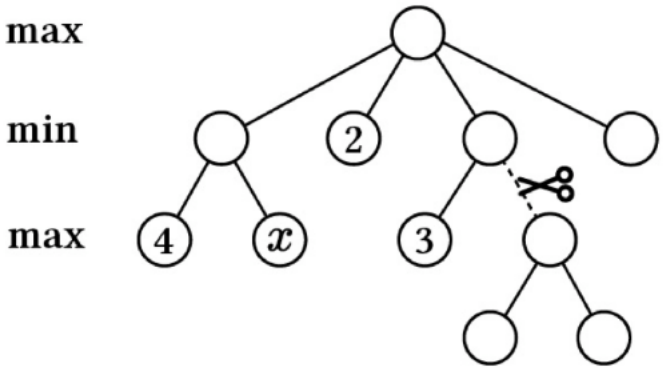
- A. B0 B1 B2 B3 B4 B5
- B. B1 B3 B5
- C. B1 B5
- D. B2 B4

评测结果 答案正确

得分 5 分

2-5 分数 5 作者 yy 单位 浙江大学

For the game tree shown below, for which values of x the dashed branch with the scissors will be pruned by α-β pruning? (Assume that the nodes are evaluated left to right)



- A. x≤2
- B. x≥3
- C. x≤6
- D. Pruning will happen for all values of x

评测结果 答案正确

得分 5 分

2-6 分数 5 作者 Yuchen Mao 单位 浙江大学

Consider the following pseudo-code.

strange($a_1, \ldots, a_n$):
1. if $n \leq 2022$ then return
2. strange($a_1, \ldots, a_{\lfloor n/2 \rfloor}$)
3. for $i = 1$ to n:
4. for $j = 1$ to $\lfloor \sqrt{n} \rfloor$:
5. print($a_i + a_j$)
6. strange($a_{\lfloor n/2+1 \rfloor}, \ldots, a_n$)

What is the running time of this pseudo-code? Your answer should be as tight as possible. (You may assume that n is a power of 2.)

- A. $O(n^{1.5})$
- B. $O(n^{1.5}) \log n$
- C. $O(n^2)$
- D. None of the other options is correct

评测结果 答案正确

得分 5 分

5-1 分数 15 Right-left Rotation 作者 陈越 单位 浙江大学

The function `RL_Rotation` is to do right-left rotation to the trouble-finder tree node `T` in an AVL tree.

```
typedef struct TNode *Tree;
struct TNode {
    int key, h;
    Tree left, right;
};

Tree RL_Rotation( Tree T )
{
    Tree K1, K2;

    K1 = T->right;
    K2 = K1->left;
    K1->left = K2 -> right ; /* 5分 */
    T->right = K2 -> left ; /* 5分 */
    K2->right = K1;
    K2 -> left = T ; /* 5分 */
    /* Update the heights */
    K1->h = maxh(Height(K1->left), Height(K1->right)) + 1;
    T->h = maxh(Height(T->left), Height(T->right)) + 1;
    K2->h = maxh(K1->h, T->h) + 1;

    return K2;
}
```

评测结果	答案正确
得分	15 分

5-2 分数 15 Inversions 作者 Ziang Qin 单位 浙江大学

The concept of inversions is significant in linear algebra. The number of inversions in a permutation p is defined as the number of pairs (i,j) such that i < j and p_i > p_j. For a given permutation p, a naive approach to figuring out the number of inversions involves using a double loop to check all possible pairs (i,j). The code for this method is as follows:

```
long long ans=0;
for(int i=1;i<n;++i)
    for(int j=i+1;j<=n;++j)
        if(p[i]>p[j]) ans+=1;
printf("%lld",ans);
```

This method has a time complexity of $O(n^2)$.
There is also a divide-and-conquer algorithm with lower time complexity ($O(n\log n)$) for finding inversions, which involves:
(1) Dividing the sequence into two halves;
(2) Counting the number of inversions within each half;
(3) Counting the number of inversions between the two halves.
The merge sort algorithm can conveniently help us count the inversions in step (3), so the process of finding inversions often accompanies the merge sort process.
Below is the code for finding inversions of a permutation using merge sort. Please complete the following program:

```
#include<stdio.h>
#define MAXN 500005
int a[MAXN],tmp[MAXN];
long long ans;
void MergeSort(int a[],int l,int r) {
    if(l==r) return;
    int mid=(l+r)/2;
    MergeSort(a,l,mid);
    MergeSort( a, mid + 1, r  3分 );
    int i=l,j=mid+1,cnt=0;
    while(i<=mid && j<=r) {
        if( a[i] > a[j]  3分 ) {
            tmp[++cnt]=a[i++];
             ans++  3分 ;
        }
        else {
             tmp[++cnt] = a[j++]  3分 ;
        }
    }
    while(i<=mid) {
        tmp[++cnt]=a[i++];
        ans+=r-mid;
    }
    while(j<=r) {
        tmp[++cnt]=a[j++];
    }
    for(int i=l;i<=r;++i) {
          3分 ;
    }
}
```

```
int read() { // Read in an integer from the standard input
and return it.
    char ch=getchar(); int x=0;
    while(ch<'0' || ch>'9') ch=getchar();
    while(ch>='0'&&ch<='9') x=(x*10)+(ch^48),
ch=getchar();
    return x;
}
int main()
{
    int n,T;
    T=read(); // There are many test cases. Each of them
includes a permutation.
    while(T--) {
        n=read(); // The length of permutation.
        for(int i=1;i<=n;++i) {
            a[i]=read();
        }
        ans=0;
        MergeSort(a,1,n); // Count the inversion number as
well as merge sorting.
        printf("%lld\n",ans);
    }
    return 0;
}
```

评测结果	部分正确
得分	6 分