

The height of an AVL tree with 19 nodes must be 5. ( We assume that the height of a single node is 1).

T

F

[答案正确: 3 分](#)

---

1-2

分数 3

作者 Yuchen Mao

单位 浙江大学

Both AVL trees and red-black trees need at most  $O(1)$  rotations per insertion.

T

F

[答案错误: 0 分](#)

---

1-3

分数 3

作者 Yuchen Mao

单位 浙江大学

Consider the B+ trees of order  $M$ . There exists a constant  $c > 0$  such that no matter how large  $M$  is, the running time of FINDKEY cannot be less than  $c \log_2 n$  where  $n$  is the number of the keys stored in the tree.

T

F

[答案正确: 3 分](#)

---

1-4

分数 3

作者 Yuchen Mao

单位 浙江大学

The longest simple path from a node  $x$  in a red-black tree to a descendant leaf has length at most twice that of the shortest simple path from node  $x$  to a descendant leaf.

T

F

[答案正确: 3 分](#)

---

1-5

分数 3

作者 Yuchen Mao

单位 浙江大学

A binomial queue with 31 keys may consists of 6 binomial trees.

T

F

[答案正确: 3 分](#)

---

1-6

分数 3

作者 Yuchen Mao

单位 浙江大学

Backtracking always performs strictly better than bruteforce.

T

F

[答案错误: 0 分](#)

---

1-7

分数 3

作者 Yuchen Mao

单位 浙江大学

Consdier the activity selection (interval scheduling) problem. In class, we have proved that an optimal solution can be obtained by selecting the activities in increasing order of thier finishing time. We can also obtain an optimal solution by selecting the activities in decreasing order of their start time.

T

F

[答案正确: 3 分](#)

---

1-8

分数 3

作者 Yuchen Mao

单位 浙江大学

Consider the recurrence  $T(n)=a^{**}T(n/b)+n^{**}d$  with  $T(1)=1$ . If  $a<b^{**}d$ , then  $T(n)$  is dominated (in asymptotic sense) by the amount work at the root of the recursion tree. If  $a>b^{**}d$ , then  $T(n)$  is dominated (in asymptotic sense) by the total amount work at the leaves of the recursion tree.

T

F

[答案正确: 3 分](#)

### 单选题

Consider an ordinary binary min-heap that supports INSERTION and DELETEMIN only. For each key  $x$  in the heap, denote its depth by  $d(x)$ . We assume that the depth of the root is 1. We define a potential function

$$\Phi(D)=c\cdot\sum_{x\in D}d(x),$$

where  $c$  is a constant that is sufficiently large. That is, the potential of a heap is proportional to the sum of the depths of the keys stored in it. Using this potential function  $\Phi$ , what are the amortized costs of INSERTION and DELETEMIN? Your bounds should be as tight as possible.

A.

$O(\log n)$  for INSERTION and  $O(\log n)$  for DELETEMIN

B.

$O(1)$  for INSERTION and  $O(\log n)$  for DELETEMIN

C.

$O(\log n)$  for INSERTION and  $O(1)$  for DELETEMIN

D.

$O(n)$  for INSERTION and  $O(n)$  for DELETEMIN

[答案正确: 5 分](#)

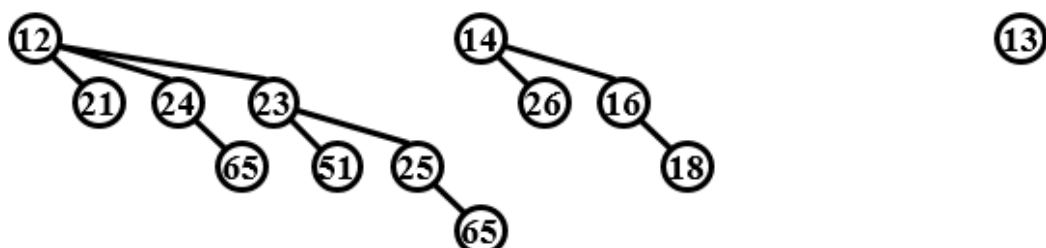
2-2

分数 5

作者 Yuchen Mao

单位 浙江大学

Delete the minimum number from the binomial queue given in the following figure. Which one of the following statements must be **FALSE**?



A.

there are two binomial trees after deletion, which are  $B_2$  and  $B_3$

B.

23 and 14 are both the roots of some binomial trees

C.

21 is the child of 13

D.

23 and 24 are siblings

[答案正确: 5 分](#)

---

2-3

分数 5

作者 Yuchen Mao

单位 浙江大学

Given the following table, what are the precision and the recall?

	Relevant	Irrelevant
Retrieved	1000	3000
Not Retrieved	7000	21000

A.

precision = 25% and recall = 12.5%

B.

precision = 12.5% and recall = 25%

C.

precision = 33.3% and recall = 14.3%

D.

precision = 14.3% and recall = 33.3%

[答案正确: 5 分](#)

---

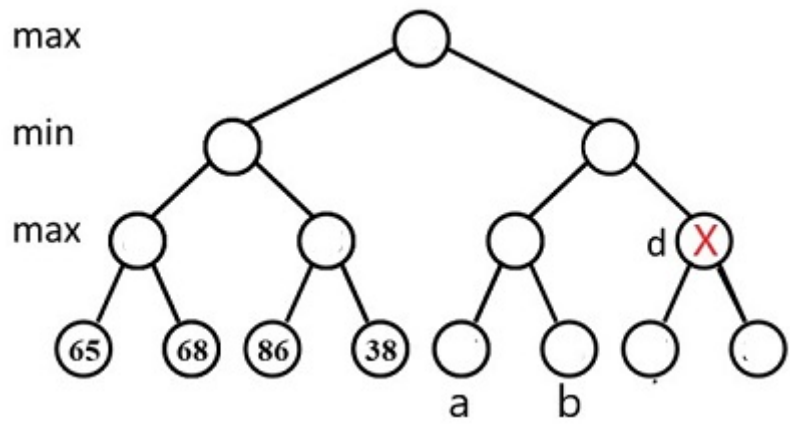
2-4

分数 5

作者 Yuchen Mao

单位 浙江大学

Consider the following game tree. If node  $d$  is pruned by  $\alpha$ - $\beta$  pruning algorithm, which of the following statements about the value of node  $a$  or node  $b$  must be correct?



- A.  
both are greater than or equal to 68
- B.  
both are less than or equal to 68
- C.  
exactly one of them is greater than 68
- D.  
exactly one of them is less than 68

[答案正确: 5 分](#)

2-5

分数 5

作者 Yuchen Mao

单位 浙江大学

Consider the following symbol frequencies for a five-symbol alphabet:

Symbol	Frequency
A	0.32
B	0.25
C	0.20
D	0.18
E	0.05

What is the average encoding length of an optimal prefix code?

- A.  
2.23
- B.  
2.31

C.

2.49

D.

2.50

[答案正确: 5 分](#)

---

2-6

分数 5

作者 Yuchen Mao

单位 浙江大学

Consider the following pseudo-code.

```
strange( $a_1, \dots, a^{**n}$ ):  
1. if  $n \leq 2022$  then return  
2. strange( $a_{\lfloor n/4+1 \rfloor}, \dots, a_{\lfloor 3n/8 \rfloor}$ )  
3. for  $i=1$  to  $n$ :  
4. print( $a^{**i}$ )  
5. strange( $a_{\lfloor 3n/4+1 \rfloor}, \dots, a^{**n}$ )
```

What is the running time of this pseudo-code? Your answer should be as tight as possible. (Hint: give a recurrence for this pseudocode and solve the recurrence with the method of your choice. You may assume  $n$  is a power of 2.)

A.

$O(1)$

B.

$O(\log n)$

C.

$O(n)$

D.

$O(n \log n)$

[答案错误: 0 分](#)

## 多选题

3-1

Let  $T$  be the splay tree that is obtained from an empty tree by the following operations: insert(1), insert(2), insert(3), insert(4), insert(5), findkey(2), findkey(3). Select all the true statements from the followings.

A.

3 is the root of  $T$ .

B.

2 is a child of 1

C.

4 is a child of 5

D.

the balanced factor of any node of  $T$  is at most 1

[部分正确: 9 分](#)

---

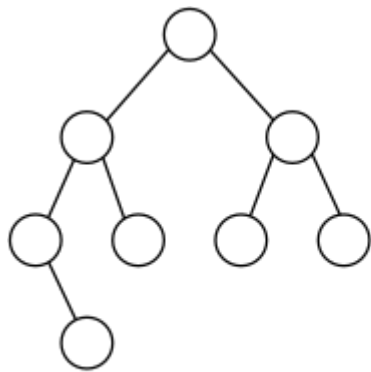
3-2

分数 6

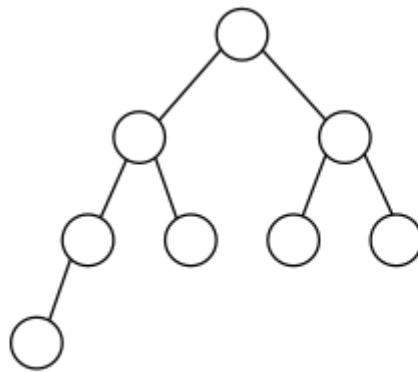
作者 Yuchen Mao

单位 浙江大学

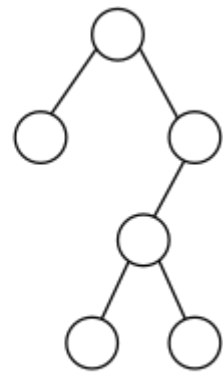
Select all the leftist heaps from the followings. Assume that the keys already satisfy the heap property.



(a)



(b)



(c)

A.

*a*

B.

*b*

C.

*c*

[部分正确: 4 分](#)

---

3-3

分数 12

作者 Yuchen Mao

单位 浙江大学

Which of the following statements about Huffman's greedy algorithm are true? Assume that the symbol frequencies sum to 1. (Choose all that apply.)

A.

A letter with frequency at least 0.4 will never be encoded with two or more bits.

B.

A letter with frequency at least 0.5 will never be encoded with two or more bits.

C.

If all symbol frequencies are less than 0.33, all symbols will be encoded with at least two bits.

D.

If all symbol frequencies are less than 0.5, all symbols will be encoded with at least two bits.

[答案正确: 12 分](#)

## 5-1

B+ Tree - Find Key

分数 16

作者 杨洋

单位 浙江大学

The function `FindKey` is to check if a given `key` is in a B+ Tree with its root pointed by `root`.

Return `true` if `key` is in the tree, or `false` if not. The B+ tree structure is defined as following:

```
static int order = DEFAULT_ORDER;
typedef struct BpTreeNode BpTreeNode;
struct BpTreeNode {
    BpTreeNode** children; /* Pointers to childrens. This field is not used by
leaf nodes. */
    ElementType* keys;
    BpTreeNode* parent;
    bool isLeaf; /* 1 if this node is a leaf, or 0 if not */
    int numKeys; /* This field is used to keep track of the number of valid
keys.
In an internal node, the number of valid pointers is always numKeys + 1. */
};
```

```
bool FindKey(BpTreeNode * const root, ElementType key){
    if (root == NULL) {
        return false;
    }
    int i = 0;
    BpTreeNode * node = root;
    while (
```

```
8 分) {
    i = 0;
    while (i < node->numKeys) {
        if (
```



```
8 分) i++;  
        else break;  
    }  
    node = node->childrens[i];  
}  
for(i = 0; i < node->numKeys; i++){  
    if(node->keys[i] == key)  
        return true;  
}  
return false;  
}
```