

A 卷

适用专业年级: 软件工程 2018 级 学号: 姓名:

考生签名:

题 号	一(30%)	二(40%)	三(20%)	四(10%)			
得 分							
卷面总分			教师签名		阅卷时间		

3. 考试结束, 请将试题纸、答题纸和草稿纸一并交给监考老师。



评阅教师	得分

提示: 在每小题列出的四个备选项中只有一个是符合题目要求的, 请将其代码写在答题纸上。错选、多选或未选均无分。

- A. $\log_2 n$
B. n^2
☒ C. 2^n
D. $2 * n$

2、本试卷审批表同试卷一并归档保存。

3. If a data element requires 4 bytes and a pointer requires 2 bytes, then a linked list representation will be more space efficient than a standard array representation when the fraction of non-zero elements is less than about: (A)

A. 2/3

B. 3/4

C. 1/3

D. 1/2

$$\text{non-zero} = n_0 \quad \text{总 } n$$

$$n_0(p+E) < nE$$

$$\frac{n_0}{n} < \frac{E}{p+E} = \frac{4}{2+4} = \frac{4}{6} = \frac{2}{3}$$

4. Which statement is not correct among the following four: (D)

☒ In a BST, the left child of any node is less than the right child, but in a heap, the left child of any node could be less than or greater than the right child.

☒ The number of empty subtrees in a non-empty binary tree is one more than the number of nodes in the tree.

☒ A general tree can be transferred to a binary tree with the root having only left child. 左孩子, 右兄弟

☒ A heap must be a full binary tree. 堆是 complete binary tree root 没有兄弟

5. The golden rule of a disk-based program design is to: (A)

☒ Minimize the number of disk accesses.

B. Eliminate the recursive calls.

C. Improve the basic operations.

D. Reduce main memory use.

6. The function of replacement selection sort is to: (B)

A. Select the maximal element.

☒ Generate the initial sorted merge files.

C. Merge the sorted file.

D. Replace some record.

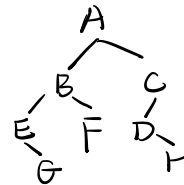
7. Assume the preorder traversal sequence of a binary tree T is ABEGFCDH, the inorder traversal sequence of T is EGBFADHC, then the postorder traversal sequence of T will be: (A)

☒ GEFBHDCA

B. EGFBHDCA

C. GEFBDHCA

D. GEBFDHCA



GEFBHDCA

8. Consider the following C++ code fragment.

```
x=191; y=200;
```

```
while(y>0)
```

```
    if(x>200)
```

```
        {x=x-10; y--;}
    else x++;
```

```
    }
```

$T(n) = O(1)$

与n无关, 为常数

- What is its asymptotic time complexity? (~~C~~) **A**
- A. $\Theta(1)$
B. $\Theta(n)$
C. $\Theta(n^2)$
D. $\Theta(n^3)$
9. When sorting n records, Selection sort will perform how many swaps in the worst case? (~~B~~)
- A. $\Theta(\log n)$
B. $\Theta(n)$
C. $\Theta(n \log n)$
D. $\Theta(n^2)$
10. The single-source shortest path problem can be used to: (~~C~~)
- A. Sort all of the graph vertices by value.
B. Sort all of the graph vertices so that each vertex is listed prior to any others that depend on it.
☒ C. Sort all of the graph vertices by distance from the source vertex.
D. None of the above.
11. In external sorting, a run is (~~A~~)
- A. A sorted sub-section for a list of records.
B. One pass through a file being sorted.
C. The external sorting process itself.
D. An external sorting method.
12. The priority queue is a structure implementing (~~C~~).
- A. inserting item only at the rear of the priority queue.
B. inserting item only at the front of the priority queue.
C. deleting item according to the priority of the item.
D. first in/first out
13. When using the weighted union rule for merging disjoint sets, the maximum depth for any node in a tree of size n will be (~~B~~)
- A. nearly constant
B. $\log n$
C. n
D. $n \log n$
14. The output from scanning a minimum heap with level traversal algorithm (~~C~~).
- A. must be an ascending sequence.
B. must be descending sequence

✓ must have a minimum item at the head position.

D. must have a minimum item at the rear position.

$n_0 = (k-1)n_k + 1$ 对满k叉树适用

15. If a binary tree has 13 nodes with two degrees and 6 nodes with one degree, how many nodes are there with zero degree? (C) ✓

A. 7

B. 15

C. 14

D. uncertain

$n_0 = n_2 + 1 = 14$ 对任意 Binary Tree 都适用

注: 不能用离散数学握手定理计算

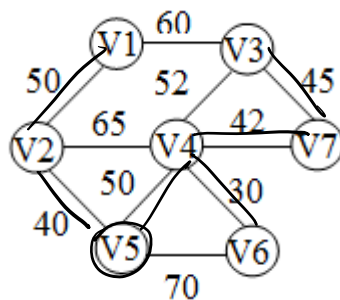
DS中 Degree 只算了出度, 没算入度

评阅教师	得分

二、应用题 (本大题共 5 小题, 每小题 8 分, 共 40 分)

提示: 有求解过程的要尽量给出解题步骤, 只有最终答案会酌情扣分。

1. List the order in which the edges of the graph in below figure are visited when running Prim's MST algorithm starting at Vertex V5. show the final MST.



2. Given sequence 43, 02, 80, 48, 26, 57, 15, 73, 21, 24, 66, select the value 57 as pivot, please write the Quicksort partition steps for pivot 57.

3. Using closed hashing, with double hashing to resolve collisions, insert the following keys into a hash table of thirteen slots (the slots are numbered 0 through 12). The hash functions to be used are H1 and H2, defined below.

Keys: 2, 8, 31, 20, 19, 18, 53, 27

$H1(k) = k \% 13$

$H2(k) = \text{Rev}(k+1) \% 11,$

Function $\text{Rev}(k)$ reverses the decimal digits of k , for example, $\text{Rev}(31)=13$, $\text{Rev}(2)=2$

Please show the hash table after all eight keys have been inserted. Be sure to indicate how you are using H1 and H2 to do the hashing.

4. You are given a series of records key/pointer pairs whose keys are: 23, 48, 12, 10, 33, 49, 20, 18,

15, 21, 45, 47, 52, 31, 28.

$m=4$

(1) Please show the B+ tree of order 4 that results from inserting these records. Assume that the leaf nodes are capable of 5 key/pointer pairs. $n=5$

(2) Show the result of deleting the record whose key is 18 from the B+ tree of (1)

5. You are given a series of values stored in an array: 10, 5, 12, 3, 2, 1, 8, 16, 29, 4

(1) Show the max-heap that results from running **buildHeap** on the above values;

(2) show the heap that results from deleting the maximum value from the max-heap you just build in (1).

评阅教师	得分

三、编程、设计及分析题（本大题共 2 小题，1 小题 8 分，2 小题 12 分，共 20 分）。

提示：每小题给出了一个程序设计要求，请按照要求写出源程序代码，如果源程序代码中出现语法错误或逻辑错误，则酌情扣分。

1. Write a function to determine whether two binary trees are the same.
2. Write a function $\text{search}(A, n, k)$ to find the $k^{\text{th}} (1 \leq k \leq n)$ smallest element in the first $n (n \geq 1)$ elements of array A. Each element in array A is distinct. For instance, $A = \{39, 18, 22, 96, 12, 62, 52, 33, 90\}$, $\text{search}(A, 6, 3)$ will return 22. What is the running time of your algorithm?

评阅教师	得分

四、分析题（本大题共 1 小题，共 10 分）。

提示：根据自己的理解和知识背景，对题目给出分析和阐述。

12 18 22

Please give examples of where the prim's algorithm and Dijkstra's algorithm are used in real life.

prim算法可以创建最经济的连通子图:

1) 应用于城市之间的交通系统

2) 应用于石油管道规划

Dijkstra算法可求最短路径:

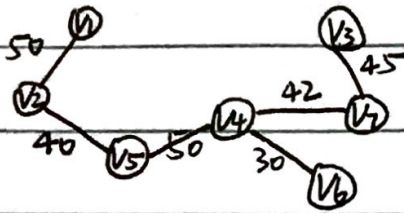
1) 应用于高德地图等APP的路线规划

2) 应用于高铁、地铁的建设

3) 应用于城市间交通系统建设

二.1. V_5-V_2 V_2-V_1 V_5-V_4 V_4-V_6 V_4-V_3 V_3-V_7

the final MST:



2. Step 1 43 02 80 48 26 66 15 73 21 24 (57)

Step 2 43 02 ²⁴~~80~~ 48 26 ⁶⁶~~24~~ 15 73 21 ⁸⁰~~66~~ 57

Step 3 43 02 ²⁴~~80~~ 48 26 ²¹~~24~~ 15 ⁷³~~21~~ ⁶⁶~~73~~ ⁸⁰~~66~~ 57

序列完成 43 02 ²⁴~~80~~ 48 26 ²¹~~24~~ 15 ⁶⁶~~73~~ ⁸⁰~~66~~ 73

3. $H_1(2) = 2$ $H_2(2) = 3 \% 11 = 3$

$H_1(8) = 8$ $H_2(8) = 9 \% 11 = 9$

$H_1(31) = 5$ $H_2(31) = 23 \% 11 = 1$

$H_1(20) = 7$ $H_2(20) = 12 \% 11 = 1$

$H_1(19) = 6$ $H_2(19) = 02 \% 11 = 2$

$\Delta H_1(18) = 5$ $H_2(18) = 91 \% 11 = 3$

$H_1(53) = 1$ $H_2(53) = 45 \% 11 = 1$

$H_1(27) = 1$ $H_2(27) = 82 \% 11 = 5$

由上, 18入槽时发生冲突 probe sequence = 5, 8, 11

27入槽时发生冲突 probe sequence = 1, 6, 11, 3.

\therefore 最终 Hash Table 如下:

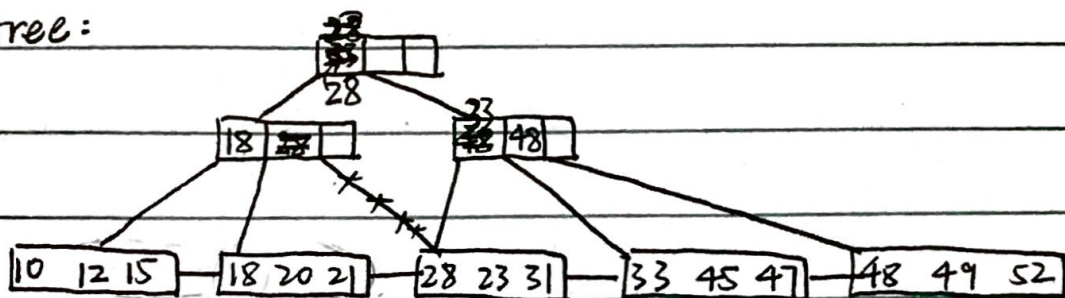
0	1	2	3	4	5	6	7	8	9	10	11	12
	53	2	27		31	19	20	8			18	

4. 内部节点 root 的 child 个数: $\lceil \frac{m}{2} \rceil \sim m$ 即 2~4

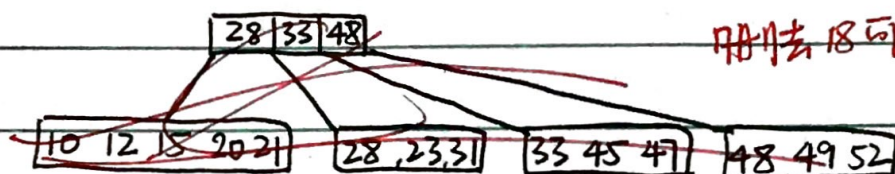
key 个数: 1~3
key

(1) 叶子节点个数: $\lceil \frac{n}{2} \rceil \sim n$ 即 3~5

B+ tree:

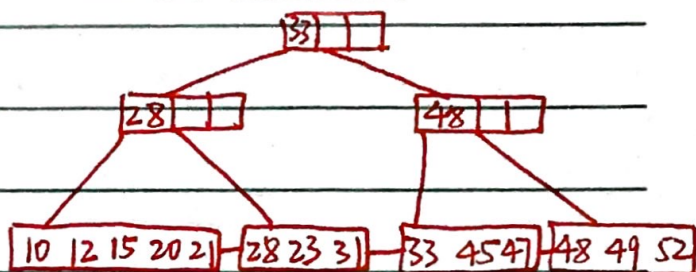
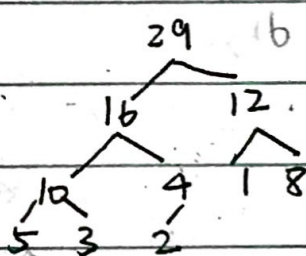


(2)

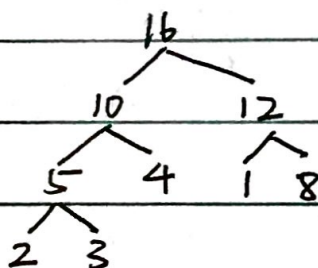


删除 18 可通过借。

5. (1)



(2)



2019A卷

三. 1. ~~bool~~ compare (Node<E> * r1, Node<E> * r2) {

if (r1 == NULL || r2 == NULL)

~~(r1->getValue() == r2->getValue())~~
if ~~(r1 == r2)~~ return ~~true~~; // 即 r1 与 r2 同为 NULL
true

else return false;

if (compare(r1->leftChild(), r2->leftChild()) == ~~false~~ ^{true}

&& compare(r1->rightChild(), r2->rightChild()) == ~~true~~ ^{true}

&& r1->getValue() == r2->getValue())

return true;

else return false;

}

~~2. int~~ ~~sed~~

2. Elem search (Elem A[], int n, int k) {

Elem B[n]; // 拷贝 A[], 用于排序 在前 n 个排序, 取第 k 个

for (int i = 0; i < n; i++) B[i] = A[i];

for (int i = 0; i < k; i++) { // 冒泡后, 选前 k 个最小在最前

for (int j = n - 1; j > i; j--)

if (A[j] < A[i]) swap(A, i, j);

}

PS: 也可以用选择排序

return B[k]; B[k-1];

}

$$T(n) = \sum_{i=0}^{k-1} (n-1-i) = \theta(kn) \quad (k \leq n)$$

void swap (Elem A[], int i, int j) {

Elem Temp = A[i];

A[i] = A[j];

A[j] = Temp;

}