1. AAACC CBACB

2. Fill the blank with correct C++ codes: (16 scores)

(1)

int r = n;             // l and r beyond array bounds

 while (l+1 != r) {     // Stop when l and r meet

    \_\_\_ int i=（l+r）/2\_\_\_\_\_;     // Look at middle of subarray

    if (K < array[i])    \_\_ r=i \_\_\_;       // In left half

    if (K == array[i])   return i ;    // Found it

    if (K > array[i])    \_\_\_ l=i \_\_\_       // In right half

 }

  // K is not in array or the greatest value is less than K

 if  K> array[0] （or l!= -1）// the lest value in the array is greater than K with l updated

then   return l  ;   // when K itself does not appear in the array

else      return  ERROR;   // the integer with the lest value greater than K

}

(2)In order to insert a new node s after the node which pointer q points to in a circular doubly linked list, we need to execute the following statements:

s->prior=q;

s->next=q->next;

q->next->prior=s ;

(3)

The height of a complete binary tree with k nodes is \_\_「log2(k+1)︱\_\_\_(1 node tree has hight 1) (2 scores)

3.

Consider a list L containing n values. L has n(n-1)/2 distinct pairs of values, each of which could potentially be in a right order or bad order. In average, that is n(n-1) /4. We can therefore say that exchange sorts are  in average amd worst case.

4.

1） 2） 3）

5.

1）



2）



3）



6.

1）pictures





2）expected bit-length



3）advantage

Huffman code scheme saves text length in most cases.

7.

1）

2）



3）

Here ， which implies we need 3 passes.

In each pass， we need read  blocks。

To read a block， we have to access disk  times.

thus, the times of reading disk is



8.

H1(22)=0, H1(41)=2, H1(53)=5, H1(46)=6, no conflict

When H1(30)=2, H2(30)=1 （2+1\*1）%11=3，so 30 enters the 3rd slot;

H1(13)=6, H2(13)=2 (6+1\*2)%11=8, so 13 enters the 8th slot;

H1(1)=3, H2(1)=8 (3+5\*8)%11= 10 so 1 enters 10 (pass by 0, 8, 5, 2 );

H1(67)=3, H2(67)=10 (3+2\*10)%11= 1 so 67 enters 1(pass by 2)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 22 | 67 | 41 | 30 |  | 53 | 46 |  | 13 |  | 1 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

9.

1）



2）



3）



10.

1） 1，6，5，2，3，4

2）

