Assignment Two

Name:	沈星宇
Student Number:	1951576

Direction:

Please answer all the questions below and hand in your answers before the due day. All work, must be handed in **on time**.

Due day:

May. 10, 2021

Questions:

1. Given an array $A = \{13, 15, 124, 28, 44, 28, 27, 5, 71\}$. Please solve following problems:
    - Arrange A in descending order by **insertion sort**.
    - Arrange A in descending order by **quick sort**.

1. (1) insertion sort:

13	15	124	28	44	28	27	5	71
15	13	124	28	44	28	27	5	71
124	15	13	28	44	28	27	5	71
124	28	15	13	44	28	27	5	71
124	44	28	15	13	28	27	5	71
124	44	28	28	15	13	27	5	71
124	44	28	28	27	15	13	5	71
124	44	28	28	27	15	13	5	71
124	71	44	28	28	27	15	13	5

(2) quick sort:

71	15	124	28	44	28	27	5	71	temp=A[0]=13
	↑i							↑j	
71	15	124	28	44	28	27	5	5	
							↑i,j		
71	15	124	28	44	28	27	13	5	
124	15	124	28	44	28	27	13	5	temp=A[0]=71
	↑i	↑j							
124	15	15	28	44	28	27	13	5	
	↑i,j								
124	71	15	28	44	28	27	13	5	
124	71	15	28	44	28	27	13	5	
124	71	27	28	44	28	27	13	5	temp=A[2]=15
			↑i			↑j			
124	71	27	28	44	28	15	13	5	
						↑i,j			
124	71	28	28	44	28	15	13	5	temp=A[2]=27
			↑i		↑j				
124	71	28	28	44	27	15	13	5	
					↑i,j				
124	71	44	28	44	27	15	13	5	temp=A[2]=28
			↑i	↑j					
124	71	44	28	28	27	15	13	5	
				↑i,j					
124	71	44	28	28	27	15	13	5	temp=A[2]=44
		↑i,j							
124	71	44	28	28	27	15	13	5	temp=A[3]=28
			↑i,j						
124	71	44	28	28	27	15	13	5	temp=A[8]=5
								↑i,j	

- Describe the basic idea of binary search for decrement arrays and give a non-recursive algorithm and also the recursive version.

(3) **a. binary search:**

二分查找对已经有序的数列进行搜索，每次将要查找的 key 与字典中间位置上的元素进行比较；如果正好相等，则检索成功；如果不等，则在对应的前半部分或后半部分继续进行检索。平均时间复杂度为 O(log n)。

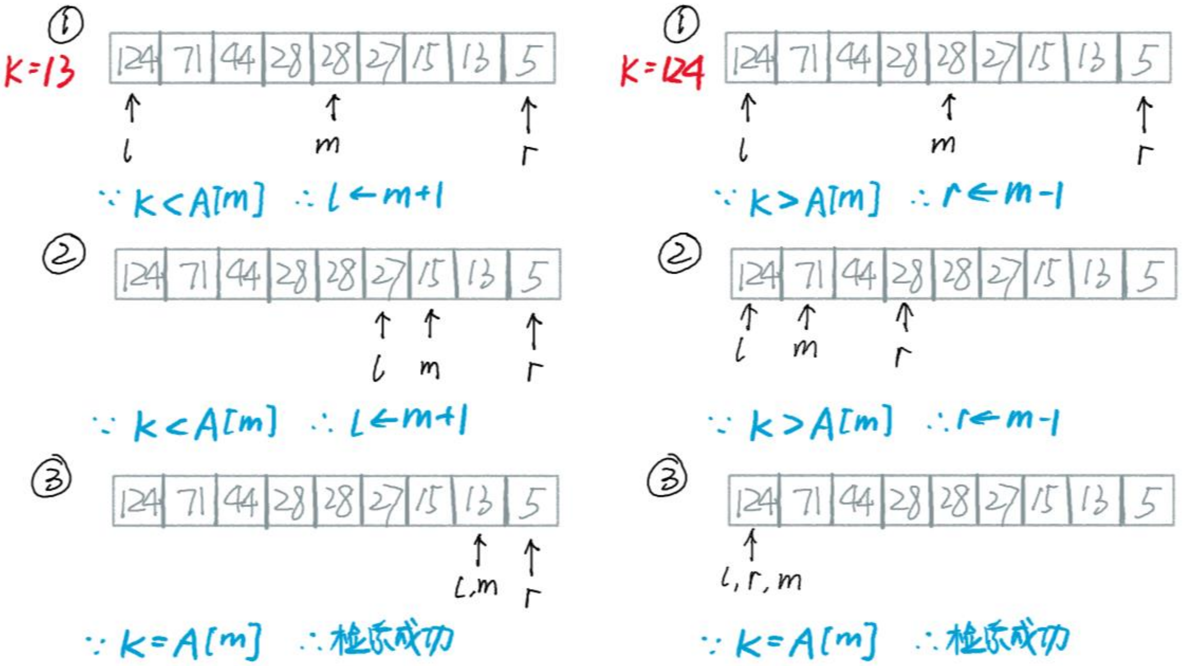
b. Non-recursive algorithm:

```
//Searches for a given value in a given decrement array by binary search
//Input: An array A[0..n-1] and a search key K
//Output: Returns the index of the first element of A that matches K or –1 if there are no matching elements
ALGORITHM  BinarySearch (A[0..n-1], K)
l ← 0; r ← n-1;
while l<=r do
    m ← $\lfloor ( l + r ) / 2 \rfloor$
    if K = A[m]
        return m
    else if K < A[m]
        l ← m + 1
    else
        r ← m - 1
return -1
```

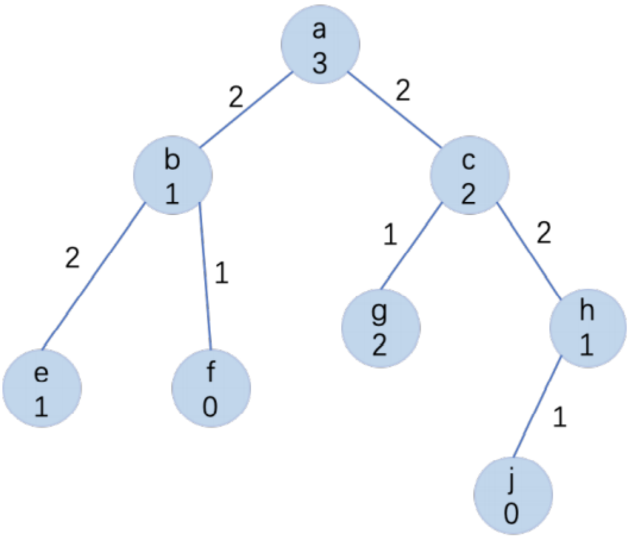
c. recursive algorithm:

```
//Searches for a given value in a given decrement array by binary search
//Input: An array A[0..n-1] and a search key K
//Output: Returns the index of the first element of A that matches K or –1 if there are no matching elements
ALGORITHM  BinarySearchRecur (A[0..n-1], l, r, K)
if l > r
    return -1
else
    m ← $\lfloor ( l + r ) / 2 \rfloor$
    if K = A[m]
        return m
    else if K<A[m]
        return BinarySearchRecur(A[0..n-1], m+1, r, K)
    else
        return BinarySearchRecur(A[0..n-1], l, m-1, K)
```

- Use above algorithms to find the elements (i.e. 13, 124) and provide necessary details of the searching process.



2. Consider the minimal cost search problem represented in the figure, where a is the start node and there are goal nodes at f and j. For each node, the heuristic cost is indicated on the node, and for each arc, the arc cost is indicated along the arc. What is the upper bound when only the start node has been explored? Which goal node is found first by Branch&Bound? What is the upper bound immediately after the first goal node is found? Is the second goal found by Branch&Bound?



(1) ∞

(2) 在分支限界法下，f 节点被先搜到

(3) 3

(4) 不是