Assignment Two

1953871 邓泉

Question 1

1. Insertion sort

初始序列 : {**13**} 15,124,28,44,28*,27,5,71

Step 1 : {**15,13**} 124,28,44,28*,27,5,71

Step 2 : {**124,15,13**} 28,44,28*,27,5,71

Step 3 : {**124,28,15,13**} 44,28*,27,5,71

Step 4 : {**124,44,28,15,13**} 28*,27,5,71

Step 5 : {**124,44,28,28*,15,13**} 27,5,71

Step 6 : {**124,44,28,28*,27,15,13**} 5,71

Step 7 : {**124,44,28,28*,27,15,13,5**} 71

Step 8 : {**124,71,44,28,28*,27,15,13,5**}

2. quick sort

初始序列 : {13,15,124,28,44,28*,27,5,71}

Step 1 : {71,15,124,28,44,28*,27,**13**,5 }　基准：13

Step 2 : {124,**71**,15,28,44,28*,27,**13,5** }　基准：71,5

Step 3 : {**124**,**71**,27,28,44,28*,**15**,**13**,**5** }　基准：124,15

Step 4 : {**124**,**71**,28*,44,28,**27**,**15**,**13**,**5** }　基准：27

Step 5 : {**124**,**71**,44,**28***,28,**27**,**15**,**13**,**5** }　基准：28*

3. 递减数组的二分搜索算法

- **非递归:**

```
// num - 递减数组；n - 数组大小；x - 搜索元素
// 若数组中存在x则返回其下标，否则返回0
int binary_search_1(int num[], int n, int x)
{
    int low = 0, high = n - 1;
    int mid;
    while (low <= high) {
        mid = (low + high) / 2;
        if (x > num[mid])
```

```
            high = mid - 1;
        else if (x < num[mid])
            low = mid + 1;
        else
            return mid;
    }
    return -1;
}
```

- **递归：**

```
// num - 递减数组；low - 起始下标；high - 终止下标；x - 搜索元素
// 若数组中存在x则返回其下标，否则返回-1
int binary_search_2(int num[], int low, int high, int x)
{
    int mid = -1;
    if (low <= high) {
        mid = (low + high) / 2;
        if (x > num[mid])
            mid = binary_search_2(num, low, mid - 1, x);
        else if (x < num[mid])
            mid = binary_search_2(num, mid + 1, high, x);
    }
    return mid;
}
```

4. 二分搜索算法搜索的过程

加粗元素 - 当前所找到的元素

- **搜索13：**

  { 124,71,44,28,**28***,27,15,13,5 } low = 0, high = 8, mid = 4

  { 124,71,44,28,28*,27,**15**,13,5 } low = 5, high = 8, mid = 6

  { 124,71,44,28,28*,27,15,**13**,5 } low = 7, high = 8, mid = 7, 返回7

- **搜索124：**

  { 124,71,44,28,**28***,27,15,13,5 } low = 0, high = 8, mid = 4

  { 124,**71**,44,28,28*,27,15,13,5 } low = 0, high = 3, mid = 1

  { **124**,71,44,28,28*,27,15,13,5 } low = 0, high = 0, mid = 0, 返回0

Question 2

(1) 当只探索到起始结点a时，上界值是正无穷 $+\infty$

(2) f 先被探索到

(3) 当找到第一个目标结点时，上界值是3

(4) 不能找到第二个目标结点

解答过程如下：

采用优先队列的分支限界法，将结点的heuristic cost 视为优先级。

设 $UB(x)$ 为探索到结点 x 时的上界，初始为 $+\infty$ ； $LB(x)$ 是探索到结点 x 时的下界，初始为 0。若当前结点 x 不是目标结点，且 $LB(x)$ 大于在它之前探索到的某一结点的 $UB(x)$ 时，对结点 x 进行剪枝操作。

- 起点a进入优先队列，$UB(a)=+\infty, LB(a)=0$。
- a结点可扩展到b、c结点，而b的优先级较高，因此按将结点b, c按顺序先后加入队列，同时将a移除， $UB(b)=+\infty, LB(b)=2$。
- b结点可以扩展到e、f两点，f的优先级更高，因此将结点f、e按顺序先后加入到优先队列中，b移除。此时队列中有c、e、f三个结点，按照优先级排列为f、e、c。
- 此时找到目标结点f，$UB(f)=2+1=3, LB(f)=2+1=3$。
- e结点不是目标结点且不可扩展，故剪去。
- 此时，队列中仅有c这一结点，将其扩展到g、h。
- g结点不是目标结点且不可扩展，故剪去。
- 对于h结点，$LB(h)=2+2=4>UB(f)=3$，根据剪枝函数应该剪去，所以不会探索到第二个目标结点j。