算法 hw4

窦嘉伟 518021911160

Q1：

三种情况分别对应 10000，100000，1000000 节点红黑树情况

序号	时间	双红
0	0.079ms	94
1	0.079ms	94
2	0.07ms	94
3	0.079ms	105
4	0.083ms	131
5	0.081ms	122
6	0.075ms	101
7	0.068ms	83
8	0.081ms	120
9	0.082ms	118

序号	时间	双红
0	1.024ms	1048
1	1.042ms	1099
2	1.995ms	1122
3	1.022ms	1045
4	1.331ms	1041
5	0.993ms	1086
6	1.007ms	1041
7	1.237ms	1117
8	2.366ms	1085
9	0.998ms	1068

序号	时间	双红
0	18.365ms	10816
1	19.023ms	10912
2	17.361ms	10962
3	17.677ms	11133
4	17.952ms	10942
5	18.149ms	10898
6	18.11ms	11071
7	18.938ms	11056
8	18.127ms	11048
9	18.332ms	11139

Q2：

1，（1，2）（4，7）（1，20）（4，15）（10，12）（29，22）（24，31）（25，27）

2,

```
Algorithm BuildKdTree(P, depth)
1.  if P contains only one point
2.    then return a leaf storing this point
3.    else if depth is even
4.           then Split P with a vertical line ℓ through the
                  median x-coordinate into P1 (left of or
                  on ℓ) and P2 (right of ℓ)
5.           else Split P with a horizontal line ℓ through
                  the median y-coordinate into P1 (below
                  or on ℓ) and P2 (above ℓ)
6.         v_left ← BuildKdTree(P1, depth + 1)
7.         v_right ← BuildKdTree(P2, depth + 1)
8.         Create a node v storing ℓ, make v_left the left
           child of v, and make v_right the right child of v.
9.         return v
```

若采用如上算法建树。通过递归函数 build 来实现对每一层的划分。每一层都需要确定中微点，其时间复杂度为 O（n），递归执行 build，由于平均划分，因此树高 logn 需要递归复杂度 O（logn），所以最终时间复杂度为 O（nlogn）；