

算法实验三

Pb18081616 谭园

一、实验要求

■实验3.1：红黑树

- 实现红黑树的基本算法，分别对整数 $n = 20, 40, 60, 80, 100$ ，随机生成 n 个互异的正整数 ($K_1, K_2, K_3, \dots, K_n$)，以这 n 个正整数作为结点的关键字，向一棵初始空的红黑树中依次插入 n 个节点，统计算法运行所需时间，画出时间曲线。
- 随机删除红黑树中 $n/4$ 个结点，统计删除操作所需时间，画出时间曲线图。

■实验3.2：区间树

- 实现区间树的基本算法，随机生成30个正整数区间，以这30个正整数区间的左端点作为关键字构建红黑树，向一棵初始空的红黑树中依次插入30个节点，然后随机选择其中3个区间进行删除。实现区间树的插入、删除、遍历和查找算法。

二、实验环境

编译环境：DEV C++

机器内存：16GB

时钟主频：2.3GHz

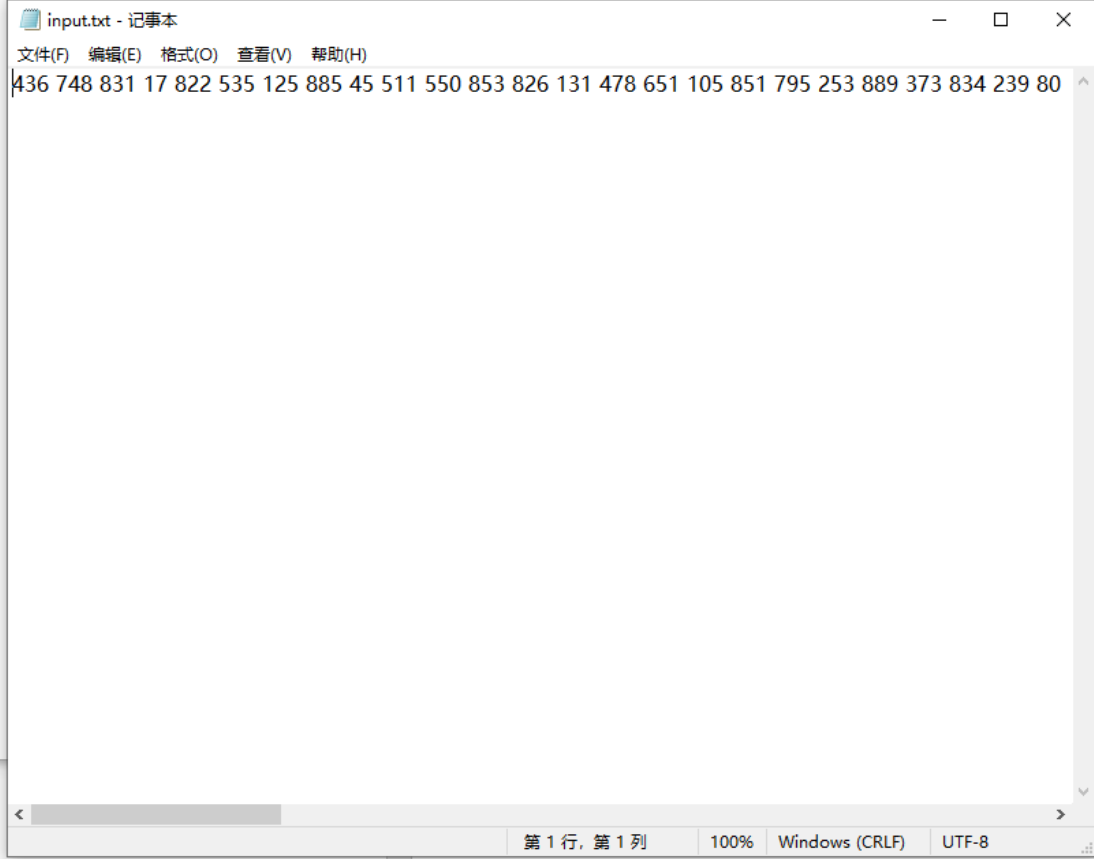
三、实验过程

- 1.利用 rand 函数生成了个随机数 (1-1000)，并写入 input 文件中。
- 2.写出各插入删除遍历查找算法的实现
- 3.得出结果并分析。

四、实验结果：单位 (us)

红黑树

Input



```
input.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
436 748 831 17 822 535 125 885 45 511 550 853 826 131 478 651 105 851 795 253 889 373 834 239 80
第 1 行, 第 1 列 100% Windows (CRLF) UTF-8
```

Output

delete_data.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
436 822 826 131 851
17 45 105 125 253 478 511 535 550 651 748 795 831 853 885
436 125 131 478 834 80 713 547 575 429
17 45 105 117 122 156 239 253 328 372 373 423 511 535 550 570 647 651 732 748 756 795 822 826 831
436 125 885 853 826 131 851 834 117 547 732 503 165 812 563
17 30 45 80 105 122 146 156 193 195 239 253 328 372 373 379 384 423 429 478 497 511 535 540 550 5
436 125 131 478 105 795 253 373 834 80 328 647 713 165 497 540 729 469 85 214
17 30 41 45 117 118 122 146 156 193 195 239 307 320 372 379 384 391 423 429 503 506 511 522 535 5
885 826 131 105 373 80 372 117 575 122 886 770 165 384 563 554 989 522 808 705 729 469 391 974 6
17 30 41 45 85 103 118 125 146 156 157 166 193 195 214 239 243 253 307 320 328 350 354 379 423 42
```

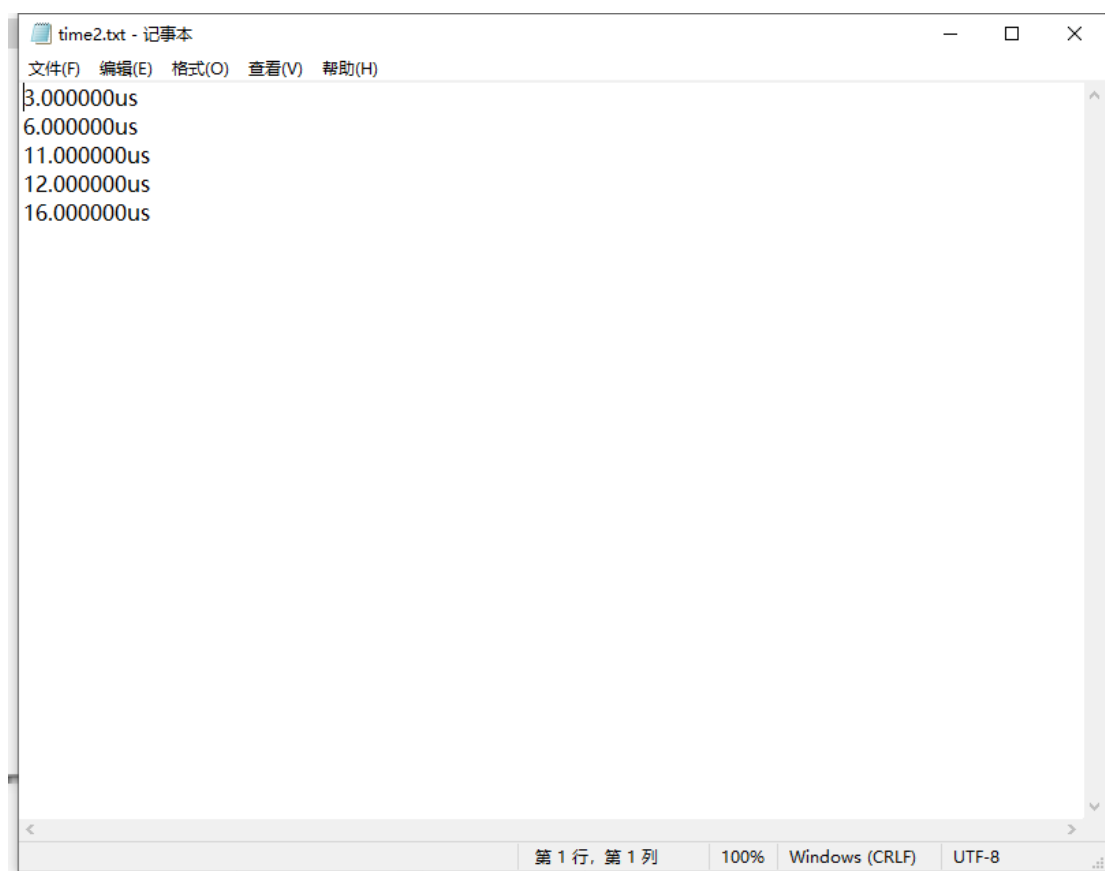
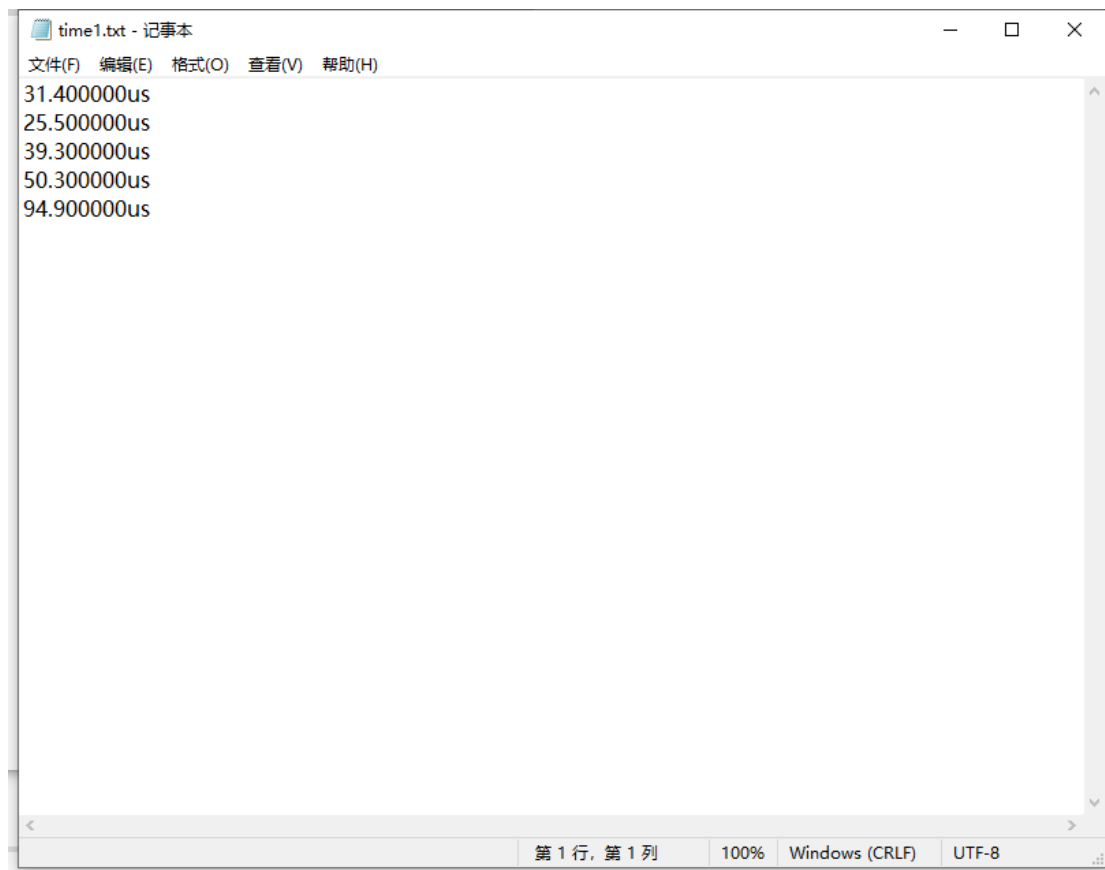
第 1 行, 第 1 列 100% Windows (CRLF) UTF-8

inorder.txt - 记事本

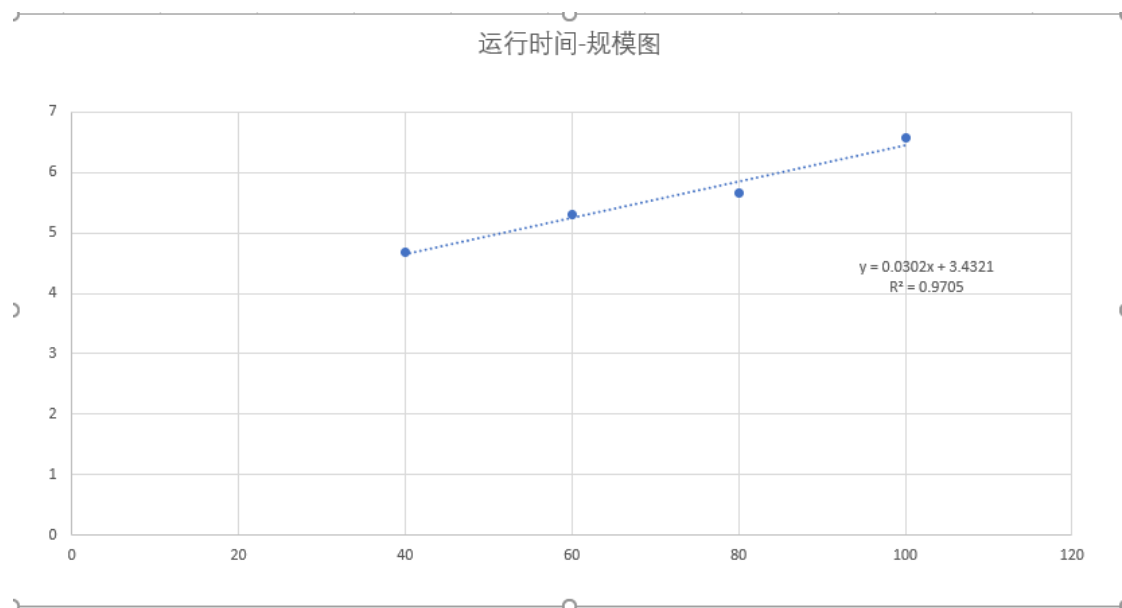
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
17 45 105 125 131 253 436 478 511 535 550 651 748 795 822 826 831 851 853 885
17 45 80 105 117 122 125 131 156 239 253 328 372 373 423 429 436 478 511 535 547 550 570 575 647
17 30 45 80 105 117 122 125 131 146 156 165 193 195 239 253 328 372 373 379 384 423 429 436 478 4
17 30 41 45 80 85 105 117 118 122 125 131 146 156 165 193 195 214 239 253 307 320 328 372 373 379
17 30 41 45 80 85 103 105 117 118 122 125 131 146 156 157 165 166 193 195 214 239 243 253 307 320
```

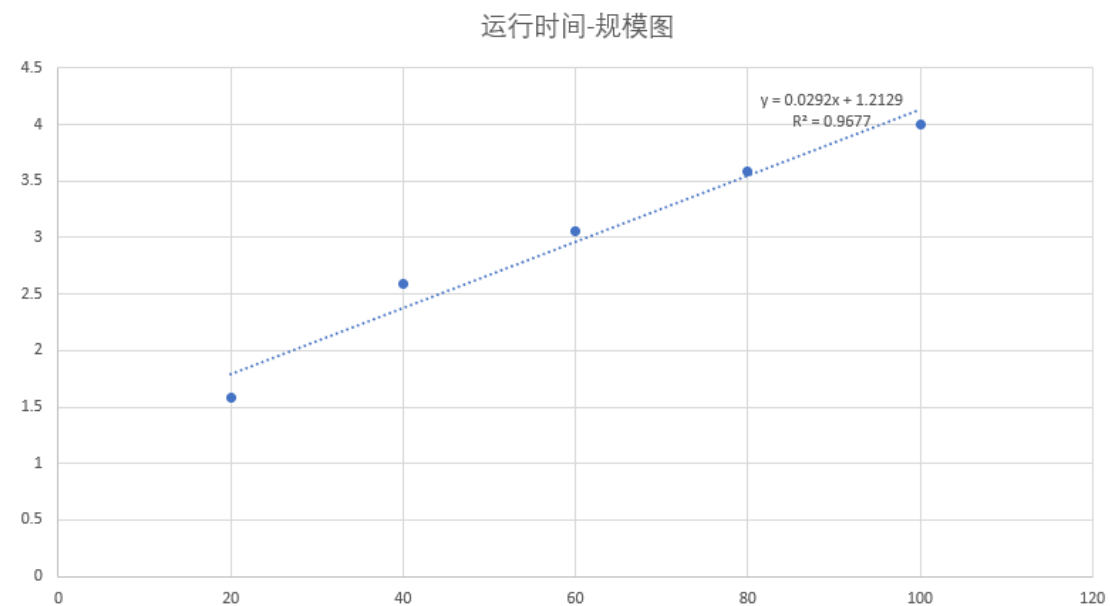
第 1 行, 第 1 列 100% Windows (CRLF) UTF-8



进行时间拟合，得到如下图。



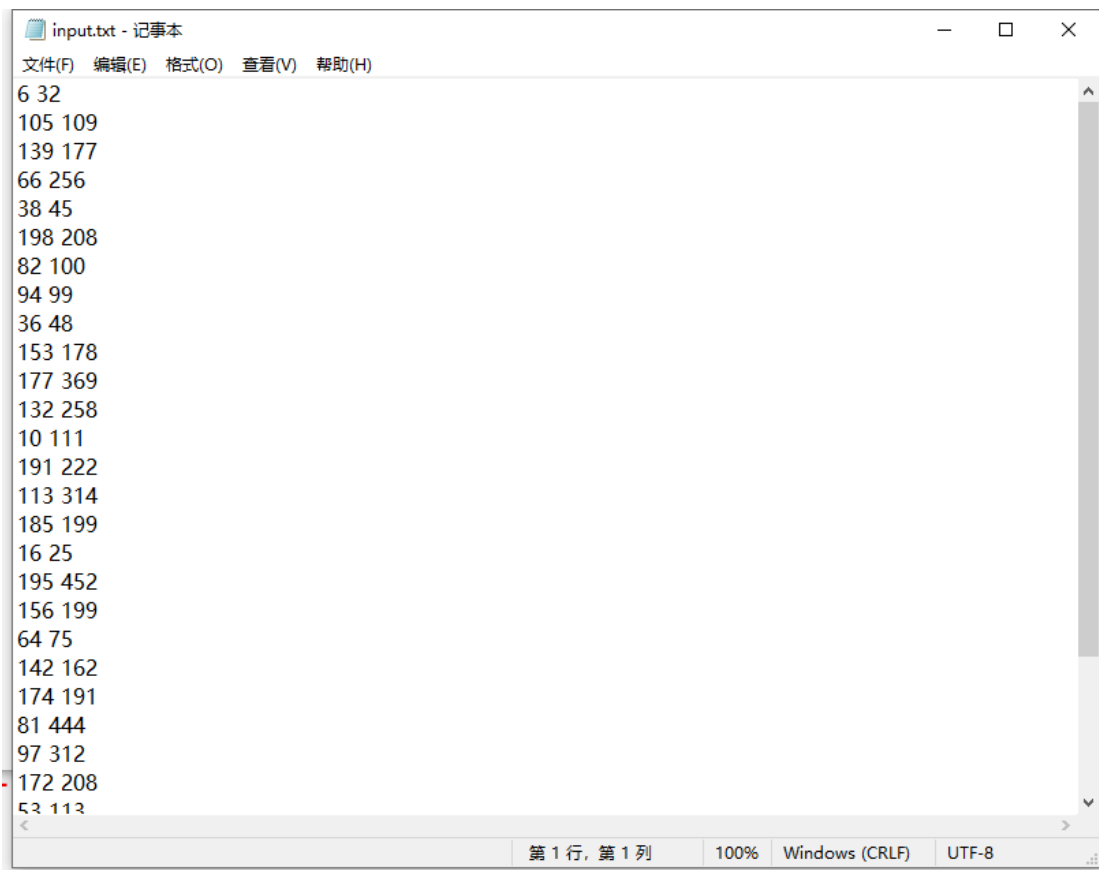
此图将运行时间做对数处理了，并舍弃了第一个规模为 20 的不太吻合的数据点。因为实验中用来计时的函数测得的时间并不单纯是排序这个过程的时间，还包括程序运行的开销。而第一个点的数据规模是 20，对实验使用的计算机来说，这样的数据量实在是过于小了，以至于时间函数本身的时间不能忽略，所以这个数据是非常不准确的。后面的四个数据可以证明这个的插入算法的时间复杂度是满足理论的。



对于删除，同样可以看出摒弃掉第一个数据点后，后面的四个数据基本符合理论时间复杂度

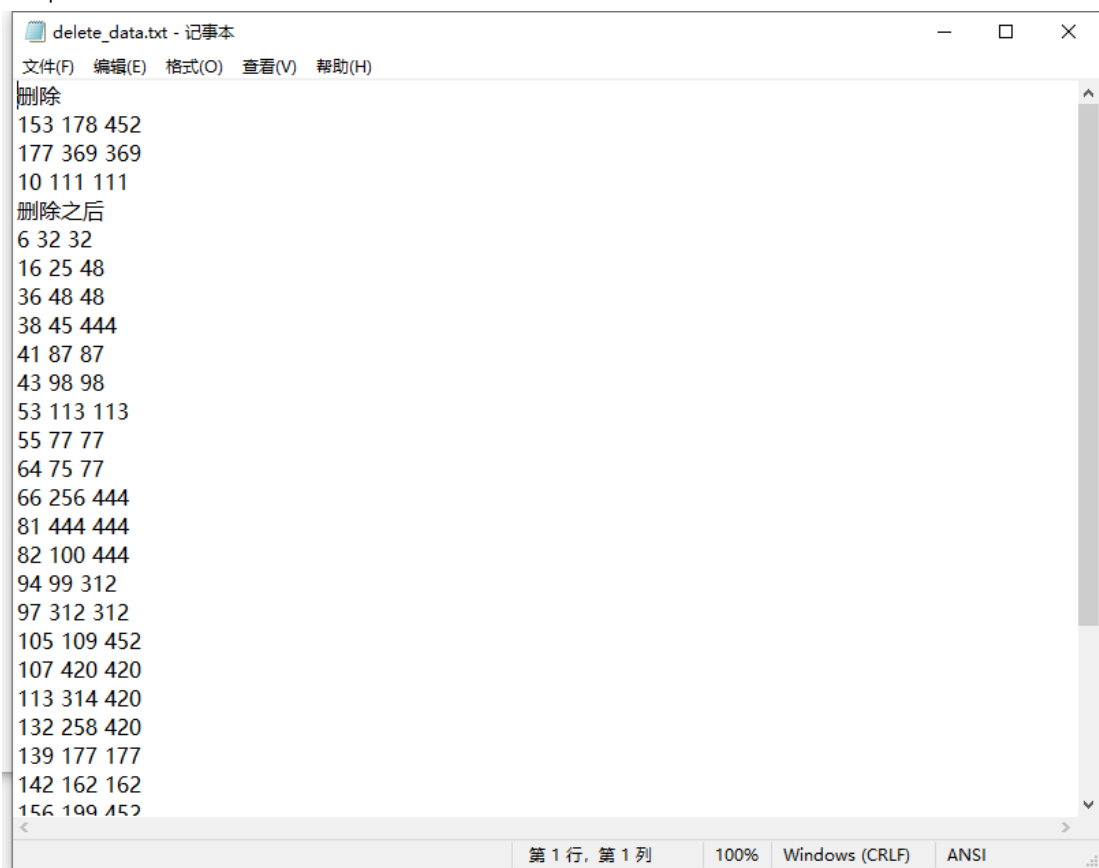
区间树

Input



```
input.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
6 32
105 109
139 177
66 256
38 45
198 208
82 100
94 99
36 48
153 178
177 369
132 258
10 111
191 222
113 314
185 199
16 25
195 452
156 199
64 75
142 162
174 191
81 444
97 312
172 208
53 113
第 1 行, 第 1 列 100% Windows (CRLF) UTF-8
```

Output



```
delete_data.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
删除
153 178 452
177 369 369
10 111 111
删除之后
6 32 32
16 25 48
36 48 48
38 45 444
41 87 87
43 98 98
53 113 113
55 77 77
64 75 77
66 256 444
81 444 444
82 100 444
94 99 312
97 312 312
105 109 452
107 420 420
113 314 420
132 258 420
139 177 177
142 162 162
156 199 452
第 1 行, 第 1 列 100% Windows (CRLF) ANSI
```

inorder.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
6 32 32
10 111 111
16 25 25
36 48 48
38 45 444
41 87 87
43 98 98
53 113 113
55 77 77
64 75 77
66 256 444
81 444 444
82 100 444
94 99 312
97 312 312
105 109 452
107 420 420
113 314 420
132 258 420
139 177 177
142 162 162
153 178 452
156 199 199
172 208 208
174 191 191
177 369 369
```

第 1 行, 第 1 列 100% Windows (CRLF) UTF-8

search.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
13 16
x=16 25

32 34
x=6 32

37 42
x=38 45
```

第 1 行, 第 1 列 100% Windows (CRLF) UTF-8

