

# web实验报告

## web 实验一

### 第一阶段

#### 主要项目文件说明

- **process\_data.py** : 根据收集的信息 ( `.csv` 文件 ) 进行进一步的数据处理, 形成以 `token` 为 `key`, `list[id]` 为值的字典, 并且压缩成二进制文件。
- **invert\_index.py** : 倒排索引表的数据结构
- **search\_prase.py** : 将输入的查询条件进行解析, 并且根据数据文件输出

#### 任务分解一

对一阶段中给定的电影和书籍数据进行预处理, 将文本表征为关键词集合

模型分词 -> 去除非中文字符 -> 过滤停用词 -> 单字选择过滤 -> 同义词合并 -> 去重

- **模型分词**: 尝试使用了 `jieba` 模型与 `pkuseg` 模型, 分词效果相近并没有很大差别, 故后续均采用 `jieba` 模型。
- **分词说明**: 分词结果不仅保留了词项同时保留有词语对应的词性信息, 通过对一个词语在句子中的词性来对多义词做一个粗糙的拟合。
- **去除非中文字符**: 有尝试保留英文信息 (将英文进行翻译) 但是调用相关 `api` 的时间过长, 所以本次实验不予考虑。
- **过滤停用词**: 将多份开源的停用词表进行了合并, 并以此为基础进行无用信息的筛出。
- **单字过滤**: 考虑到即使使用了停用词表仍然存在部分意义不明的词语, 我们考虑删除最可能是无用信息的一部分子集——为数词、量词、代词、介词等非关键词性的单字
- **同义词合并**: 没有找到特别合适的同义此表故放弃

#### 任务分解二

基于前一阶段形成的分词结果, 在经过预处理的数据集上建立倒排索引表, 并以合适的方式存储生成的倒排索引文件

在前面的环节我们已经能够得到以 `tuple[token,pos]` (词汇, 词性) 为 `key`, `list[id]` (评论 `id`) 为值的字典

倒排索引表为继承 `UserList` 的子类，以下是会额外设计用于查询的成员函数

## 倒排索引表的数据结构

- 考虑到跳表的指针开销，以及一层的跳表实际上减少的时间开销实际有限。本实验中用索引的增长来模拟跳表：既没有额外的空间开销，又在时间开销上最优可以达到  $O(\log n)$

### quick\_and

(and one other)，对one与other两个排序列表(不妨设为从小到大的排序)做交集

以索引的大幅度增长来模拟跳表(维护一个跳转长度)

对于one的索引我们有(other对称，不再赘述)：

如果one的当前值小于other当前值(大于情况对称，不再赘述)：

快速增长：每次将跳转长度增长一倍，直到找到一个大于等于other的值

在快速增长的查询阶段发生以下事件作对应的处理

事件一：找到一个大于other的值确定了区间，二分法找到other的值，给索引赋值，跳转长度减半

事件二：等于other的值，将结果放入result，跳转长度减半

事件三：超出one的范围，跳转长度调整为one的长度减去上一个索引，跳转长度减半

- 空间复杂度：与朴素算法一致
- 时间复杂度：最优可以达到  $O(\log n)$  这由索引的指数级增长容易得到，对于较坏的情况我们去考虑相较于朴素算法的额外的损耗（在本算法中为跳转失败后的：额外比较 1 + 二分查找  $\log x$  - 本身就需要比较 1），但是也要考虑如果进行了二分查找那么至少有收益约为  $x - 1 - \log x$  这个可由  $\frac{x}{2} - 1 + \frac{x}{4} - 1 + \dots$  直到  $\frac{x}{2^n} < 1$  得到，因此综合考虑最坏的情况比较次数大约为  $\frac{4}{3}n$  这个完全可以接受，实际情况中几乎无法达到（ps:本实验中的事件设置与快速增长为对 tcp 拥塞控制的模仿不代表这是最优的方案，实际上对完全随机的两个列表因该都可以让最坏的复杂度达到  $O(n)$  以上）

### 实际测试(循环 100 次)

```
"(and 花 婚姻 感情 生活)"
```

```
查询时间:{'quick_and': 0.005951881408691406}
```

```
查询时间:{'and_op': 0.026388168334960938}
```

```
"(and 记录 大学 爱情)"
```

```
查询时间:{'quick_and': 0.020933151245117188}
```

```
查询时间:{'and_op': 0.03333687782287598}
```

```
"(and 生活 早晨 美好)"
```

```
查询时间:{'quick_and': 0.0011165142059326172}
```

```
查询时间:{'and_op': 0.0015380382537841797}
```

可以看到本实验的算法 `quick_and` 明显优于原先的朴素算法 `and_op` 但是由于本次实验数据量较小索引表普遍长度 $\leq 1000$  导致没有更进一步体现出算法的优势，当然这里的测试尽量选取了出现可能较多的 `token` 否则因为数据量较小，导致可能会没有该 `token` 影响实验比较效果。

### or\_op

对于多个有序列表的合并，使用基于最小堆的算法，该算法较为常见在此不多做赘述

### sub\_op

对一个集合删去他的子集

### and\_sub\_op

当考虑到数量级较大的评论数量，对 `not a` 的计算作 `a` 的补集是不合适的，我们考虑如若 `not a` 以上还有 `and` 运算符时进行综合计算来回避补集无法计算的问题，如 `(and a (not b))`，我们只需计算 `a-b` 在这个函数我们通过 `sub_op(a,quick(a,b))` 来实现

## 任务分解三

优化你生成的倒排索引表，对于给定的包含任意组合（如括号）的布尔查询（例如 (动作 and 剧情)or(科幻 and not 恐怖)），使其能够支持复杂的布尔查询操作。返回符合查询规则的电影或/和书籍 集合并以合适的方式展现给用户（例如输出 tag 等）

## 查询语句

查询模板：句子 @ 查询条件

- **句子** -> 分词 -> 将所有查询到的分词倒排索引表作交集  
ps:我们允许句子中的 `(token, pos)` 未出现在先前生成的字典中，来模拟一个模糊搜索的效果
- **查询条件**：精准查找，必须出现或者必须不在

### 查询语法：

采用类似 `lisp` 的语法 `(op arg1 arg2 ...)`，支持 `and`，`or`，`not` 的操作符

ps:减少大量在生成语法树上的工作

### 查询语句的优化：

- **flatten**: 将尽量减少语法树的高度，让 `and` 操作有更多的选择空间，让 `or` 的合并更加高效

```
(and a (and b c)) -> (and a b c)
(or a (or b c)) -> (or a b c)
```

- **not\_down**: 把 `not` 操作符下沉到语法树的最底层，简化后续对不可计算补集问题的解决

```

(or
  (not
    (and
      a
      (and b c)))
  (or e f))

=> not_down => flatten =>

(or
  (not 'a')
  (not 'b')
  (not 'c')
  'e'
  'f')

```

- **\_and\_on\_not**: 对语法树节点进行标记, 若以该节点为根的子树是可计算的则标记为 `True`, 否则为 `False`, 并且遵循下列规则

```

a => a.tag=True
(not a) => (not a).tag=False
(and a b c ...) => any(a.tag,b.tag,c.tag,...)
(or a b c ...) => all(a.tag,b.tag,c.tag,...)

```

- **refuse\_search**: 若根节点是不可计算的则抛出错误, 要求用户进一步明晰条件
- **and\_on\_not**: 该函数做一个转换, 将可计算的语法树的所有不可计算的叶子节点 (即(not a) 这样的否命题) 与一个可计算的节点 `b` 结合如: `(and b (not a))` 这样就可以使用在上一节提到的 `and_sub_op` 来计算。
  - 对于原子命题直接返回倒排索引表
  - 对于 `(or a b ...)` 分别计算 `a b ...` 最后返回 `or_op(a,b,...)`
  - 对于 `(and a b ...)`
    - 如果所有节点都是可计算的, 那么分别计算后简单的返回 `quick_and(a,b,...)`
    - 如果存在节点是不可计算的, 将所有可计算的节点用 `and` 合并 如 `(and a b c d)`, 其中 `c,d` 是可计算的有转换 `(and a b (and c d))` (值得注意的一点是在这一步中我们会将 `(and c d)` 标记在他计算后会存入缓存, 这是因为接下来的变换中, 这个式子可能会重复出现) 依次把不可计算的节点与可计算的节点结合成可计算的节点, 那么式子转换为 `(and b (and a (and c d)))`

但是这样依旧不够当不可计算的节点是否命题时当然可以使用 `and_sub_op` 来计算，但是对于不可计算的复合命题我们需要进行进一步的处理，接下来我们只考虑 `(and a b)` 其中 `a` 是不可计算的复合命题，`b` 是可计算的。

- 当 `a.op==and` 时执行 `and_on_not(flatten((and a b)))` 这样的递归是一定会结束的，因为每次传入的语法树高都在减小
- 当 `a.op==or` 时作变化，不妨设 `a=(or c,d)` (更多参数的情况类似) `(and (or c,d) b)`  
`=> (or and_on_not((and c b)) and_on_not(and d b))`

## 计算

做完以上优化后该语法树可以进入计算 (不可计算，直接返回要求用户修改查询)

- 计算的原则较为简单，相应的节点使用 `quick_and or_op and_sub_op` 即可，前期的处理已经保证了该计算一定能够进行

## (and a b c...)的优化：

- 事实一：长度较小的到排序表合并耗时更少
- 事实二：相关词汇同时出现在同一篇文章中的概率会更大

## 算法处理：

- 先考虑那些是原子命题的合并，对于这些原子命题两两计算相似性 (在本次实验中将相似性分为 9 个等级其中相应的相似性字典在 `dict_similar.txt` 中)，每次将最不相似的一对 `token` (设为 `a, b`) 且他们指向的倒排序表长度和最小的两个倒排序表作交集得到倒排序表 `H` (如果倒排序表指向的是同一个，这个基于地址的比较复杂度仍保持  $O(n)$ )，则跳过)，并把 `a, b` 的倒排序表指向 `H`，最后所有的命题指向同一个结果

## 任务分解四

任选两种课程中介绍过的索引压缩方法加以实现，如按块存储、前端编码等，

- 文件 `id` 采用变长数字：每八位中的第一位作为是否延续的标志位，储存在 `id_list.bin`
- 索引 `token` 按块储存，解压根据索引长度解析，值保存 `id_list` 开始的二进制文件地址与长度，在运行时按需解析并实现有缓存机制

直接将整个数据结构序列化花费空间：1788KB

`id_list.bin` 倒排索引表储存花费：227KB

`token2id_map.bin` `token` 的存储花费：303KB

## 实现效果

运行结果截图

Open: [屏幕截图 2024-11-10 233515.png](#)

```
(python3123) PS D:\code\exp\web_exp\web1> python -u "d:\code\exp\web_exp\web1\search_prase.py"
请输入查询语句:生活 @ (and 花 (or (not 婚姻) (not 感情)))
查询语句:生活,附加查询条件:(and 花 (or (not 婚姻) (not 感情)))
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\yanhx\AppData\Local\Temp\jieba.cache
Loading model cost 0.735 seconds.
Prefix dict has been built successfully.
附加条件后最终结果:[0, 1, 5, 7, 20, 23, 25, 32, 60, 71, 91, 93, 98, 132, 209, 216, 224, 309, 358, 412, 525, 526, 535, 844, 1019]
```

## 日志输出

```
请输入查询语句:生活 @ (and 花 (or (not 婚姻) (not 感情)))
查询语句:生活,附加查询条件:(and 花 (or (not 婚姻) (not 感情)))
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\yanhx\AppData\Local\Temp\jieba.cache
Loading model cost 0.682 seconds.
Prefix dict has been built successfully.
找到('生活', 'vn')相关词条
词条相关:[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34, 35, 36, 37, 39, 40,
41, 42, 43, 45, 46, 50, 51, 52, 53, 54, 55, 56, 59, 60, 61, 62, 65, 66, 67,
69, 71, 76, 77, 78, 79, 80, 81, 82, 84, 87, 88, 90, 91, 92, 93, 94, 95, 97,
98, 99, 100, 102, 103, 104, 105, 106, 107, 108, 109, 112, 113, 114, 115, 116,
117, 118, 121, 123, 124, 125, 127, 128, 129, 130, 131, 132, 133, 136, 137,
138, 140, 141, 142, 143, 144, 146, 147, 149, 150, 151, 152, 153, 155, 156,
157, 158, 159, 160, 162, 163, 164, 165, 166, 167, 168, 169, 170, 173, 174,
177, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192,
193, 196, 197, 198, 199, 200, 201, 203, 204, 205, 207, 208, 209, 210, 212,
214, 215, 216, 217, 220, 221, 222, 223, 224, 225, 226, 229, 230, 231, 232,
233, 234, 237, 238, 241, 242, 244, 246, 247, 248, 249, 250, 252, 253, 256,
257, 258, 259, 263, 264, 265, 266, 267, 268, 269, 270, 272, 273, 274, 275,
276, 277, 279, 281, 284, 285, 287, 288, 289, 291, 297, 301, 302, 303, 304,
306, 308, 309, 310, 311, 313, 315, 316, 317, 318, 319, 321, 325, 328, 329,
330, 331, 332, 334, 335, 341, 342, 343, 344, 346, 347, 350, 356, 357, 358,
360, 361, 363, 364, 365, 366, 368, 371, 373, 374, 376, 378, 380, 381, 383,
385, 387, 388, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402,
404, 405, 410, 412, 413, 414, 415, 416, 418, 421, 424, 425, 429, 433, 436,
437, 440, 442, 443, 444, 445, 446, 447, 448, 450, 452, 454, 455, 456, 459,
460, 461, 462, 466, 467, 468, 469, 470, 471, 473, 475, 476, 478, 479, 481,
485, 491, 492, 499, 501, 502, 506, 507, 508, 509, 510, 512, 513, 514, 522,
525, 526, 527, 528, 530, 534, 535, 536, 538, 539, 542, 545, 547, 551, 552,
554, 556, 557, 559, 562, 563, 564, 566, 567, 568, 571, 572, 574, 575, 578,
579, 580, 583, 587, 588, 591, 593, 596, 598, 599, 600, 601, 602, 603, 604,
```

607, 612, 621, 625, 632, 639, 643, 648, 650, 654, 656, 658, 667, 668, 674,  
677, 678, 679, 680, 681, 685, 686, 689, 690, 696, 697, 705, 706, 707, 710,  
711, 715, 716, 722, 723, 726, 727, 733, 734, 735, 738, 741, 749, 750, 752,  
755, 756, 760, 761, 762, 763, 767, 770, 771, 772, 773, 774, 775, 776, 778,  
779, 780, 785, 786, 787, 788, 793, 797, 806, 808, 812, 813, 814, 819, 820,  
823, 830, 831, 833, 834, 837, 841, 842, 843, 844, 849, 856, 857, 859, 860,  
862, 874, 876, 879, 881, 882, 885, 886, 892, 894, 895, 897, 900, 903, 905,  
908, 909, 910, 911, 914, 915, 916, 918, 921, 926, 927, 930, 934, 935, 938,  
940, 941, 942, 944, 947, 949, 950, 964, 965, 968, 969, 970, 972, 974, 975,  
980, 981, 982, 989, 991, 997, 999, 1010, 1011, 1013, 1017, 1019, 1021, 1026,  
1028, 1030, 1031, 1033, 1045, 1046, 1051, 1054, 1057, 1058, 1060, 1064, 1066,  
1075, 1076, 1081, 1084, 1089, 1091, 1093, 1095, 1097, 1101, 1104, 1107, 1109,  
1110, 1111, 1112, 1114, 1115, 1116, 1124, 1125, 1126, 1127, 1131, 1137, 1142,  
1146, 1149, 1151, 1153, 1155, 1157, 1158, 1161, 1164, 1165, 1170, 1173, 1175,  
1179, 1183, 1184, 1188, 1190, 1193, 1195]

(and 生活) 查询语句结果: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,  
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34, 35,  
36, 37, 39, 40, 41, 42, 43, 45, 46, 50, 51, 52, 53, 54, 55, 56, 59, 60, 61,  
62, 65, 66, 67, 69, 71, 76, 77, 78, 79, 80, 81, 82, 84, 87, 88, 90, 91, 92,  
93, 94, 95, 97, 98, 99, 100, 102, 103, 104, 105, 106, 107, 108, 109, 112, 113,  
114, 115, 116, 117, 118, 121, 123, 124, 125, 127, 128, 129, 130, 131, 132,  
133, 136, 137, 138, 140, 141, 142, 143, 144, 146, 147, 149, 150, 151, 152,  
153, 155, 156, 157, 158, 159, 160, 162, 163, 164, 165, 166, 167, 168, 169,  
170, 173, 174, 177, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189,  
190, 191, 192, 193, 196, 197, 198, 199, 200, 201, 203, 204, 205, 207, 208,  
209, 210, 212, 214, 215, 216, 217, 220, 221, 222, 223, 224, 225, 226, 229,  
230, 231, 232, 233, 234, 237, 238, 241, 242, 244, 246, 247, 248, 249, 250,  
252, 253, 256, 257, 258, 259, 263, 264, 265, 266, 267, 268, 269, 270, 272,  
273, 274, 275, 276, 277, 279, 281, 284, 285, 287, 288, 289, 291, 297, 301,  
302, 303, 304, 306, 308, 309, 310, 311, 313, 315, 316, 317, 318, 319, 321,  
325, 328, 329, 330, 331, 332, 334, 335, 341, 342, 343, 344, 346, 347, 350,  
356, 357, 358, 360, 361, 363, 364, 365, 366, 368, 371, 373, 374, 376, 378,  
380, 381, 383, 385, 387, 388, 391, 392, 393, 394, 395, 396, 397, 398, 399,  
400, 401, 402, 404, 405, 410, 412, 413, 414, 415, 416, 418, 421, 424, 425,  
429, 433, 436, 437, 440, 442, 443, 444, 445, 446, 447, 448, 450, 452, 454,  
455, 456, 459, 460, 461, 462, 466, 467, 468, 469, 470, 471, 473, 475, 476,  
478, 479, 481, 485, 491, 492, 499, 501, 502, 506, 507, 508, 509, 510, 512,  
513, 514, 522, 525, 526, 527, 528, 530, 534, 535, 536, 538, 539, 542, 545,  
547, 551, 552, 554, 556, 557, 559, 562, 563, 564, 566, 567, 568, 571, 572,  
574, 575, 578, 579, 580, 583, 587, 588, 591, 593, 596, 598, 599, 600, 601,

602, 603, 604, 607, 612, 621, 625, 632, 639, 643, 648, 650, 654, 656, 658, 667, 668, 674, 677, 678, 679, 680, 681, 685, 686, 689, 690, 696, 697, 705, 706, 707, 710, 711, 715, 716, 722, 723, 726, 727, 733, 734, 735, 738, 741, 749, 750, 752, 755, 756, 760, 761, 762, 763, 767, 770, 771, 772, 773, 774, 775, 776, 778, 779, 780, 785, 786, 787, 788, 793, 797, 806, 808, 812, 813, 814, 819, 820, 823, 830, 831, 833, 834, 837, 841, 842, 843, 844, 849, 856, 857, 859, 860, 862, 874, 876, 879, 881, 882, 885, 886, 892, 894, 895, 897, 900, 903, 905, 908, 909, 910, 911, 914, 915, 916, 918, 921, 926, 927, 930, 934, 935, 938, 940, 941, 942, 944, 947, 949, 950, 964, 965, 968, 969, 970, 972, 974, 975, 980, 981, 982, 989, 991, 997, 999, 1010, 1011, 1013, 1017, 1019, 1021, 1026, 1028, 1030, 1031, 1033, 1045, 1046, 1051, 1054, 1057, 1058, 1060, 1064, 1066, 1075, 1076, 1081, 1084, 1089, 1091, 1093, 1095, 1097, 1101, 1104, 1107, 1109, 1110, 1111, 1112, 1114, 1115, 1116, 1124, 1125, 1126, 1127, 1131, 1137, 1142, 1146, 1149, 1151, 1153, 1155, 1157, 1158, 1161, 1164, 1165, 1170, 1173, 1175, 1179, 1183, 1184, 1188, 1190, 1193, 1195]

找到('花', 'v')相关词条

词条相关:[0, 1, 5, 7, 20, 23, 25, 33, 60, 71, 81, 93, 96, 98, 132, 148, 209, 224, 236, 240, 251, 309, 358, 412, 438, 449, 451, 488, 525, 526, 535, 590, 655, 665, 682, 688, 693, 704, 708, 769, 791, 799, 827, 844, 870, 883, 890, 923, 952, 971, 979, 1049, 1071, 1113, 1177, 1191, 1197]

找到('花', 'n')相关词条

词条相关:[32, 91, 96, 119, 216, 896, 1019, 1027]

花 查询语句结果:[0, 1, 5, 7, 20, 23, 25, 32, 33, 60, 71, 81, 91, 93, 96, 98, 119, 132, 148, 209, 216, 224, 236, 240, 251, 309, 358, 412, 438, 449, 451, 488, 525, 526, 535, 590, 655, 665, 682, 688, 693, 704, 708, 769, 791, 799, 827, 844, 870, 883, 890, 896, 923, 952, 971, 979, 1019, 1027, 1049, 1071, 1113, 1177, 1191, 1197]

找到('婚姻', 'n')相关词条

词条相关:[2, 3, 11, 18, 23, 35, 69, 81, 150, 163, 237, 245, 410, 414, 447, 499, 554, 587, 589, 599, 652, 720, 839, 1017, 1071]

婚姻 查询语句结果:[2, 3, 11, 18, 23, 35, 69, 81, 150, 163, 237, 245, 410, 414, 447, 499, 554, 587, 589, 599, 652, 720, 839, 1017, 1071]

(and '花' (not '婚姻')) 查询语句结果:[0, 1, 5, 7, 20, 25, 32, 33, 60, 71, 91, 93, 96, 98, 119, 132, 148, 209, 216, 224, 236, 240, 251, 309, 358, 412, 438, 449, 451, 488, 525, 526, 535, 590, 655, 665, 682, 688, 693, 704, 708, 769, 791, 799, 827, 844, 870, 883, 890, 896, 923, 952, 971, 979, 1019, 1027, 1049, 1113, 1177, 1191, 1197]

找到('感情', 'n')相关词条

词条相关:[0, 3, 5, 8, 9, 10, 12, 13, 17, 24, 25, 31, 64, 68, 81, 106, 109, 163, 189, 200, 211, 217, 245, 267, 276, 283, 284, 307, 320, 337, 346, 381, 429,



446, 464, 467, 511, 537, 554, 590, 734, 744, 844, 990, 991, 997, 1008, 1026, 1030, 1172, 1173]

感情 查询语句结果:[0, 3, 5, 8, 9, 10, 12, 13, 17, 24, 25, 31, 64, 68, 81, 106, 109, 163, 189, 200, 211, 217, 245, 267, 276, 283, 284, 307, 320, 337, 346, 381, 429, 446, 464, 467, 511, 537, 554, 590, 734, 744, 844, 990, 991, 997, 1008, 1026, 1030, 1172, 1173]

(and '花' (not '感情')) 查询语句结果:[1, 7, 20, 23, 32, 33, 60, 71, 91, 93, 96, 98, 119, 132, 148, 209, 216, 224, 236, 240, 251, 309, 358, 412, 438, 449, 451, 488, 525, 526, 535, 655, 665, 682, 688, 693, 704, 708, 769, 791, 799, 827, 870, 883, 890, 896, 923, 952, 971, 979, 1019, 1027, 1049, 1071, 1113, 1177, 1191, 1197]

(or (and '花' (not '婚姻')) (and '花' (not '感情')) 查询语句结果:[0, 1, 5, 7, 20, 23, 25, 32, 33, 60, 71, 91, 93, 96, 98, 119, 132, 148, 209, 216, 224, 236, 240, 251, 309, 358, 412, 438, 449, 451, 488, 525, 526, 535, 590, 655, 665, 682, 688, 693, 704, 708, 769, 791, 799, 827, 844, 870, 883, 890, 896, 923, 952, 971, 979, 1019, 1027, 1049, 1071, 1113, 1177, 1191, 1197]

附加条件后最终结果:[0, 1, 5, 7, 20, 23, 25, 32, 60, 71, 91, 93, 98, 132, 209, 216, 224, 309, 358, 412, 525, 526, 535, 844, 1019]