Hw7

1. 顺序/逆序下，数组规模的影响

Q = 90，Size选取100，1000，10000，100000四种大小

顺序的数据集是1—size

乱序使用`random_shuffle`函数打乱

结果：

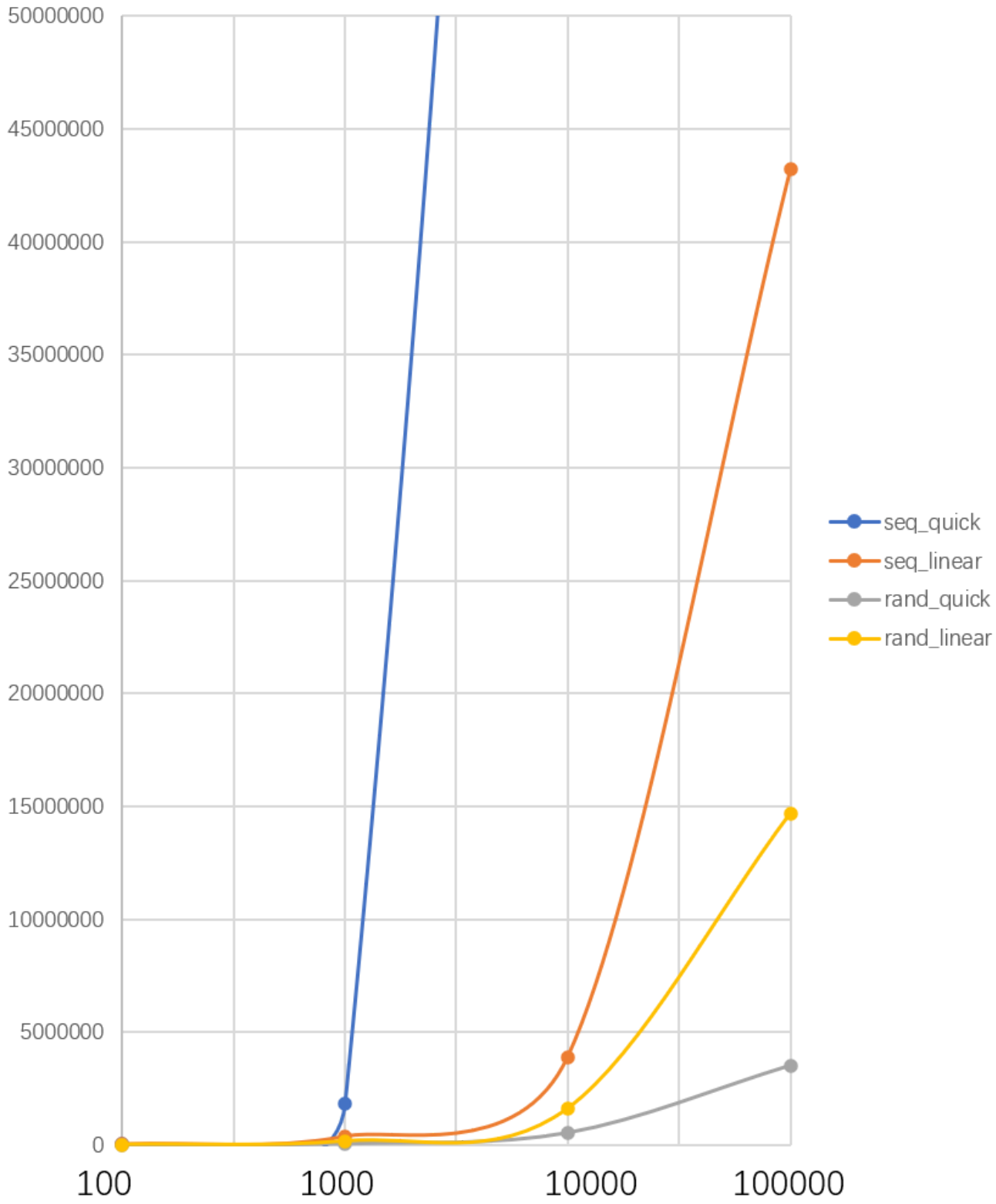
```
Sequence Case: n = 100
LinearSelect: res = 50  time cost =24064
QuickSelect: res = 50  time cost =18728
Random Case: n = 100
LinearSelect: res = 50  time cost =13440
QuickSelect: res = 50  time cost =5802

Sequence Case: n = 1000
LinearSelect: res = 500  time cost =369164
QuickSelect: res = 500  time cost =1847484
Random Case: n = 1000
LinearSelect: res = 500  time cost =162048
QuickSelect: res = 500  time cost =48236


Sequence Case: n = 10000
LinearSelect: res = 5000  time cost =3896120
QuickSelect: res = 5000  time cost =191387074
Random Case: n = 10000
LinearSelect: res = 5000  time cost =1609678
QuickSelect: res = 5000  time cost =531042

Sequence Case: n = 100000
LinearSelect: res = 50000  time cost =43189013
QuickSelect: res = 50000  time cost =18006429828
Random Case: n = 100000
LinearSelect: res = 50000  time cost =14690574
QuickSelect: res = 50000  time cost =3520530
```

结果：

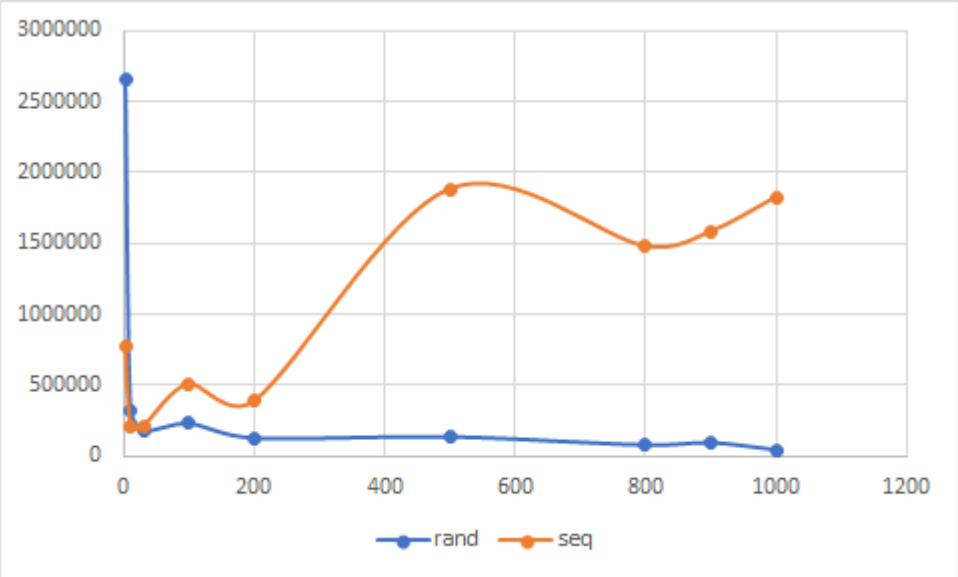


- QuickSelect在乱序表现最好
- QuickSelect最坏的情况是顺序：$O(n^2)$
- LinearSelect的表现相对稳定，由于是基于QuickSort的，所以乱序比顺序好一些。

2. 探究Q的影响

选取size = 1000。Q的值依次为2，10，50，100，200，500，800，900，1001

Q	rand	seq
2	2659838	767606
10	326142	210954
30	178540	212616
100	228964	508234
200	126344	386758
500	134796	1884602
800	77366	1479910
900	94972	1583326
1001	43780	1826680



- 如果选择的Q值太小，则会导致算法需要递归较多次才能得到结果；如果选择的Q值太大，则会导致每组数据的大小增加，需要更多的内存空间存储每组
- 对于乱序集来说，Q越大，越接近QuickSelect，而QuickSelecet在乱序集中表现较好，所以Q越大效果越好
- 对于顺序集来说，Q = 30时效果较好，此时顺序乱序集合执行速度差不多，更好的解决了QuickSelect的最坏情况。

其余Size大小中，效果最好的Q的选择（以顺序集结果为选择标准）如下

Size	Q
100	10
1000	30
10000	90
100000	100

- Q的选择是综合考虑了顺序和乱序执行情况下，以两者执行速度差不多且较快为选择依据
- 在数据较小时，Q选取为$\sqrt{Size}$ 较好，数据大后，基本在90、100左右效果较好