

实验思路

数据集

分别创建数据集规模为 100, 500, 1000, 2000, 3000 的有序、无序数据集
有序数据集：为 0-size 的顺序数据
无序数据集：使用 rand()随机生成

实验方法

对于每种规模的数据集，分别采用 Linear Select（分组规模为 5）和 Quick Select 方法，查找秩为 0-size 的所有数据，统计总的查找时间，并进行对比。
将 Linear Select 算法中数据的分组规模分别调整为 5、10、15、20、25，对比在不同数据规模下的运行时间。

实验结果

测试结果

比较 Linear Select（分组规模为 5）和 Quick Select 方法

Size	100	500	1000	2000	3000
Linear select/o	0.01	0.143	0.535	1.969	3.639
Linear select/u	0.009	0.144	0.519	1.788	4.049
Quick select/o	0.001	0.004	0.027	0.062	0.211
Quick select/u	0.001	0.004	0.027	0.068	0.208

注：o 代表数据为有序，u 代表数据为无序，下同

比较 Linear Select 在不同分组规模下的性能

Size	100	500	1000	2000	3000
Q=5/o	0.011	0.157	0.48	1.994	3.659
Q=5/u	0.01	0.147	0.606	2.037	4.421
Q=10/o	0.006	0.094	0.285	1.129	2.162
Q=10/u	0.005	0.083	0.402	1.076	2.292
Q=15/o	0.005	0.075	0.235	0.791	1.706
Q=15/u	0.005	0.07	0.338	0.865	2.003
Q=20/o	0.005	0.073	0.239	1.067	1.878
Q=20/u	0.005	0.073	0.269	1.002	2.101
Q=25/o	0.005	0.062	0.228	0.906	1.657
Q=25/u	0.004	0.069	0.245	0.927	2.106

结果分析

1. Quick select 算法的性能远优于 Linear Select
2. Linear Select 对于有序数据的运算时间优于无序数据，且随着数据规模的增大，差距越来越明显；Quick select 对于有序数据和无序数据的运算时间近似。
其原因可能是 Linear Select 需要大量使用排序算法，对于有序数据，会显著降低这部分算法的时间消耗。
3. Linear Select 算法随着分组规模的增大，其性能也逐渐提升，但当分组规模越来越大时，其提升的性能也越来越少且存在上限。