

## 2 亿数据量下测试效率：

算法	耗时
Gpu_quicksort	1500ms
Cpu_quicksort	780ms
Gpu_quickfind	70ms
Gpu_radixsort	82ms
Gpu_radixfind	12ms

说明:前缀为 cpu 的是没有经过 cuda 并行优化,只通过 cpu 运算的;  
后缀为 sort 的表示对数据全排序, 后缀为 find 的表示只查找数据中的 top\_k 数据; 中缀表示使用算法名称。

## 效率对比说明：

在所有测试中, 只寻找 top\_k 的算法显然优于对所有数据的全排列算法。而 radixfind 算法用时为 quickfind 算法的 1/6 左右, 原因是 quickfind 效率一般情况下一次能筛掉一半左右的数据。但 radixfind 算法一次排序是对 8 位 2 进制位排序, 在数据稀疏且分布均匀的情况下, 可以认为  $2^8$  (256) 个桶中的数据量是一样大的, 理论上当 top\_k 较小时, 一次可以筛到只剩 1/256 左右数据; 在数据分布稠密时, 一次也能筛到只剩 1/10-1/20 左右的数据, 所以效率比 quickfind 更高。

## 编译：

Nvcc 文件名 -arch=sm\_35 -rdc=true