**服务计算前沿技术实验报告**

学号： 1190201421

姓名： 张 瑞

**目录**

[第1章 算法实现正确性分析 3](#_Toc78202844)

[第2章 QoS数据集的产生 4](#_Toc78202845)

[第3章 结果与运算速度分析 4](#_Toc78202846)

[第4章 实验结果启示 4](#_Toc78202847)

## 第1章 算法实现正确性分析

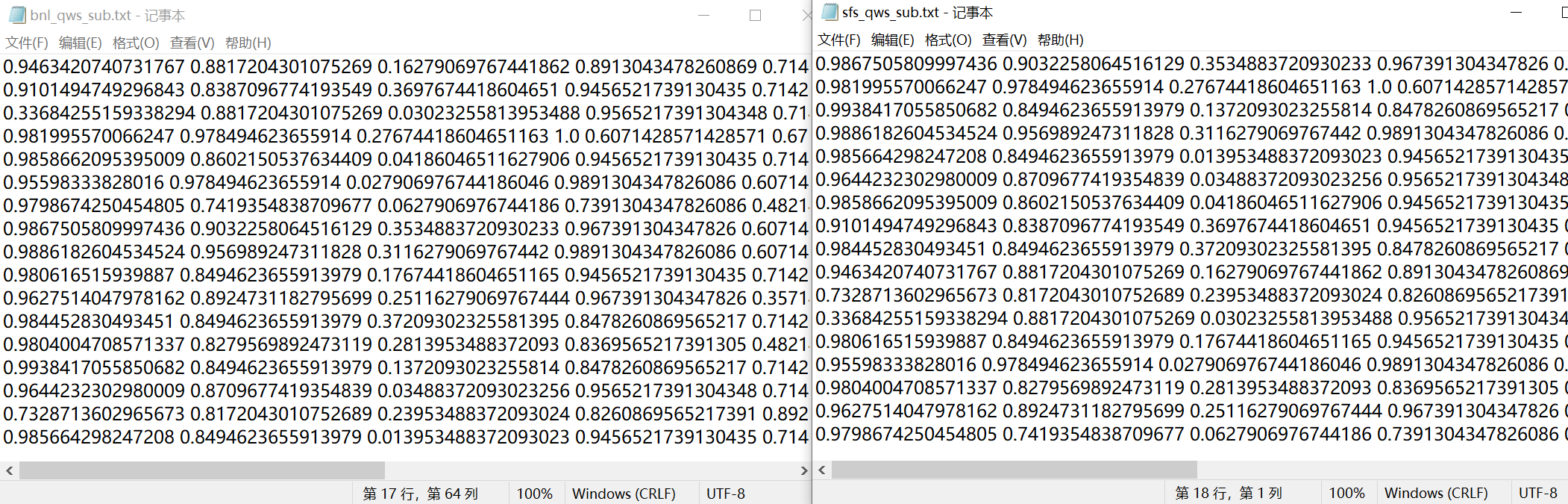
BNL是一种简单暴力的算法，通过块嵌套循环的方式，判断任意两个点是否存在支配关系。SFS是一种改进的BNL算法，首先根据单调偏好函数对整个数据集进行排序，然后在此基础上按BNL计算。

在QWS数据集上，分别用qws\_sub.txt和qws\_normal.txt验证正确性。将BNL算法和SFS算法的结果分别保存在两个txt文件中，观察二者结果发现：两个算法对相同txt的处理结果是一样的。

在qws\_normal.txt上验证正确性时，BNL与SFS最终的结果均筛选出226个向量：

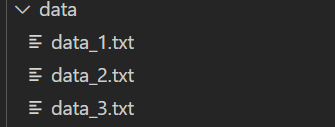


在qws\_sub.txt上验证正确性时，BNL与SFS最终的结果均筛选出17个向量：



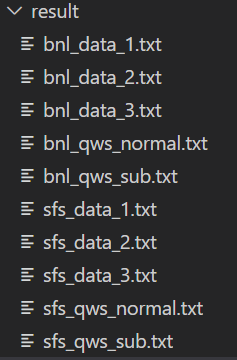
## 第2章 QoS数据集的产生

在仿真工具randdataset上产生了3个QoS数据集：data\_1包含3k\*10维的独立数据、3k\*10维的正相关数据以及4k\*10维的反相关数据；data\_2包含6k\*10维的独立数据、6k\*10维的正相关数据以及8k\*10维的反相关数据；data\_3包含16k\*10维的独立数据、16k\*10维的正相关数据以及18k\*10维的反相关数据。



## 第3章 结果与运算速度分析

全部运行结果被写入以下文件：



下面进行运算速度分析，首先查看各个txt分别在BNL和SFS算法下的运算时间。

data\_1.txt 10K\*10：



data\_2.txt 20K\*10：



data\_3.txt 50K\*10：



观察运行时间可以发现，SFS相较BNL有明显的提速。

## 第4章 实验结果启示

为了更好地提高速度，应该使用更巧妙的经过优化的算法。前面所实现的BNL算法就是一个最基础的算法，实现起来简单但是运行速度欠佳，SFS算法作为对BNL算法的一个改进，明显提高了运算速率。除了SFS算法以外，还有SCAN扫描算法，D&C算法，NN算法，BBS算法等。对数据进行预处理也是一种提速的办法，其实SFS算法就是先对数据进行了排序，在排序基础上再使用BNL算法，提速效果明显。