# ANLY550 Homework 3

Siyao Peng (sp1184) & Yilun Zhu (yz565)

## 1 Analytically

Here is the analytical calculation to find the crossover point for strassen versus naive multiplication.

We assume that at some $n_0$, for all lower-end multiplications, we will have naive multiplication in either case. We need to compare one step higher (the last recursion layer) where $2n_0$ is broken down to 4 matrices of size $n_0$.

By naive multiplication, we know that there are $n_0^3$ multiplications and $n_0^2(n_0 - 1)$ additions. So at time T($n_0$) we have

$$T(n_0) = n_0^3 - n_0^2(n_0 - 1) = 2n_0^3 - n_0^2 \tag{1}$$

Now, the task is to let $T_{strassen}(2n_0) < T_{naive}(2n_0)$. We know from induction that

$$T_{naive}(2n_0) = 2(2n_0)^3 - (2n_0)^2 = 16n_0^3 - 4n_0^2 \tag{2}$$

Based on the pseudocode we learned in class and the code provided in *strassen.py*, we know there are 18 additions at $T_{strassen}(2n_0)$ and 7 calls of naive multiplication of size $n_0$. So we know that

$$T_{strassen}(2n_0) = 7T(n_0) + 18(n_0)^2 = 7(2n_0^3 - n_0^2) + 18(n_0)^2 = 14n_0^3 + 11n_0^2 \tag{3}$$

By letting $T_{strassen}(2n_0) < T_{naive}(2n_0)$, we have:

$$14n_0^3 + 11n_0^2 < 16n_0^3 - 4n_0^2 \tag{4}$$

$$2n_0^3 - 15_0^2 > 0 \tag{5}$$

$$n_0^2(2n_0 - 15) > 0 \tag{6}$$

$$\therefore n_0 > 15/2 = 7.5 \tag{7}$$

We conclude that analytically the cross-over point is around 7.5 if we only consider arithmatic operations and all other operations are free.

# 2 Empirically

We manually define the cross-over point = 19, based on empirical results below.

When we run mode="4", we loop through matrices of dimension 128*128 to 299*299, the distribution of crossover points is shown in the following figure where the blue line shows a value = 19 closer to the mean.

The distribution of crossover point is also shown below.

| Min | 1st Quart. | Median | Mean | 3rd Quart. | Max |
|---|---|---|---|---|---|
| 8 | 14 | 17 | 18.38 | 22 | 44 |

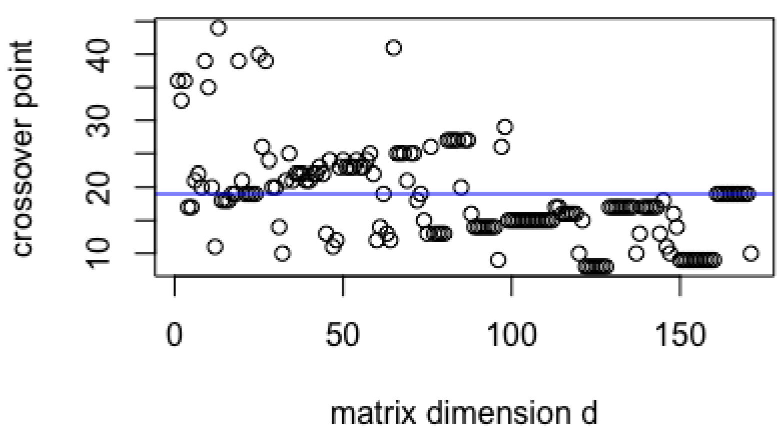


Figure 1: Plot of crossover point versus matrix dimension d

## 2.1 Documentation

To reproduce the result of finding the optimal cross-over point, run:

```
python.py 3 dimension inputfile
```

The algorithm will find the optimal edge point between $[4, 100)$ for the given dimension.

If you want to see the range of optimal cross-over points, please run:

```
python.py 4 x inputfile
```

The algorithm ignores the input dimension and selects dimensions from 128 to 299. The range of optimal edge is between $[4, 100)$ as well. The algorithm will also save all possible optimal cross-over points into a file named "`optimal`".

### 2.2 Find the optimal cross-over point

Assume the algorithm uses conventional matrix multiplication throughout the loop. If we want to efficiently calculate matrix multiplication, the algorithm will stop using conventional algorithm at some cross-over point $n_0$. In other words, the algorithm will start using Strassen's algorithm when the empirical overall runtime is shorter than the runtime of conventional algorithm.

We implement the algorithm as follows:

1. Given the dimension in the command, the algorithm calculates the runtime of multiplying two matrices in naive multiplication. The runtime is $t_0$.

2. We test all possible optimal points between $[4, 100)$, according to the analytical result in (7). For each loop, the runtime is $T = t_1, t_2, ..., t_n$

3. For the first time we encounter the case such that $t_i \in T$ is smaller than $t_0$, the optimal cross-over point is found.

**Correctness:**

Suppose the cross-over point is $n_0$. Conventional algorithm is used for both naive and Strassen's algorithms for all $n_i < n_0$. Because the conventional algorithm runs in $O(n^3)$ and Strassen's algorithm runs in $O(n^{\log_2 7})$ when $n$ is large enough, using Strassen's empirically at some point will decrease the runtime. In this case, the first time the Strassen's algorithm is faster, the cross-over point is found in the loop.[1]

[1] Empirically, the runtime of Strassen's algorithm is gradient descenting when the possible edge $n_i$ increases. However, suppose the dimension is 129, the algorithm will never run naive multiplication for $48 * 48$ matrices $(129, 65, 33, 17...)$. Though on our computers

# 3 Comparison between analytical and empirical results

It is always the case the empirical cross-over point is higher than the analytical result. Because in Strassen's algorithm, we do not calculate the runtime of dividing one $n$ matrix into 4 $n/2$ matrices, padding for odd-size matrices, and initializing results to be zeros of proper dimensions. Therefore, Strassen's would be slower than the optimal case and the empirical cross-over point is bigger than the analytical result.

$n_0 = 48$ is faster than $n_0 = 33$. Therefore, we only count $n_0$ as the cross-over point when the first time Strassen's is faster than conventional algorithm.