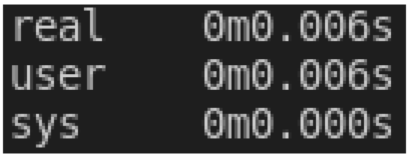
# VE477 Lab3 Report

## Question 2

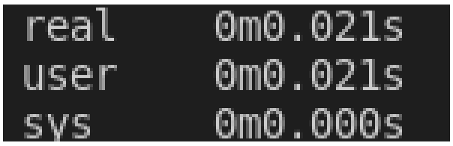
## Question 3

#### 3.1 Sortcount

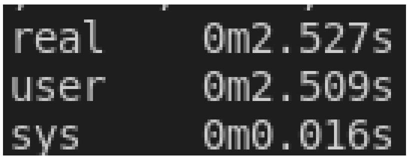
Three different input size, which contain `100`, `1000` and `10000` are chosen. And the corresponding time cost for those three testcases are shown in the following figures



```
real	0m0.006s
user	0m0.006s
sys	0m0.000s
```



```
real	0m0.021s
user	0m0.021s
sys	0m0.000s
```



```
real	0m2.527s
user	0m2.509s
sys	0m0.016s
```

Notice that the time complexity for this algorithm can be calculated by master theorem,

$$\mathcal{T}(n) = 2\mathcal{T}(\frac{n}{2}) + \mathcal{O}(n \log n)$$

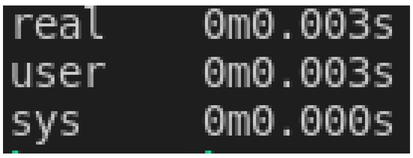
which gives

$$\mathcal{O}(n \log n)$$

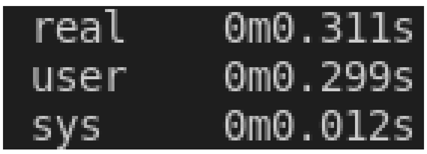
Referring to the time table provided in the lecture, which shows `0.6`, `10` and `130` disregarding the unit. We can see that the ratio are basically matched.

#### 3.2 Gale-Shapley

Three different input size, which contain `100`,`1000` and `2000` are chosen. And the corresponding time cost for those three testcases are shown in the following figures



```
real	0m0.003s
user	0m0.003s
sys	0m0.000s
```



```
real	0m0.311s
user	0m0.299s
sys	0m0.012s
```

```
real    0m1.234s
user    0m1.186s
sys     0m0.048s
```

During the lecture, we proved that the time complexity for `Gale-Shapley` is

$$\mathcal{O}(n^2)$$

Referring to the time table provided in the lecture, which shows `10`, `1000` for input `100` and `1000` respectively without unit. We can see the ratio is about 100 times, which matches the real result well.

**3.3 Knapsack**

Two different input size, which contain `100` and `1000` are chosen. And the corresponding time cost for those tesecases are shown in the following figures

```
real    0m0.031s
user    0m0.027s
sys     0m0.004s
```

```
real    0m3.780s
user    0m3.776s
sys     0m0.004s
```

The time complexity for this algorithm is

$$\mathcal{O}(NM)$$

where `M,N` are the total weight and the number of times respectively. In these input, we can view the time complexity as

$$\mathcal{O}(n^2)$$

We see that the real time cost is similar to what we have discussed previously in 3.2.