# ran_walk_diagram

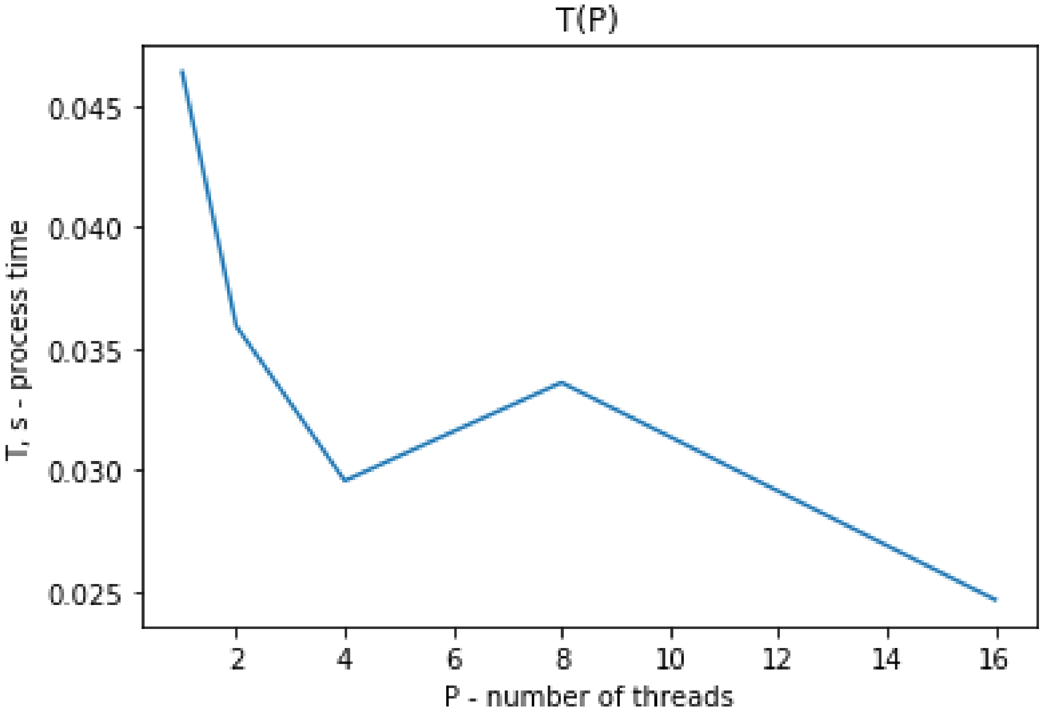
November 29, 2017

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
```

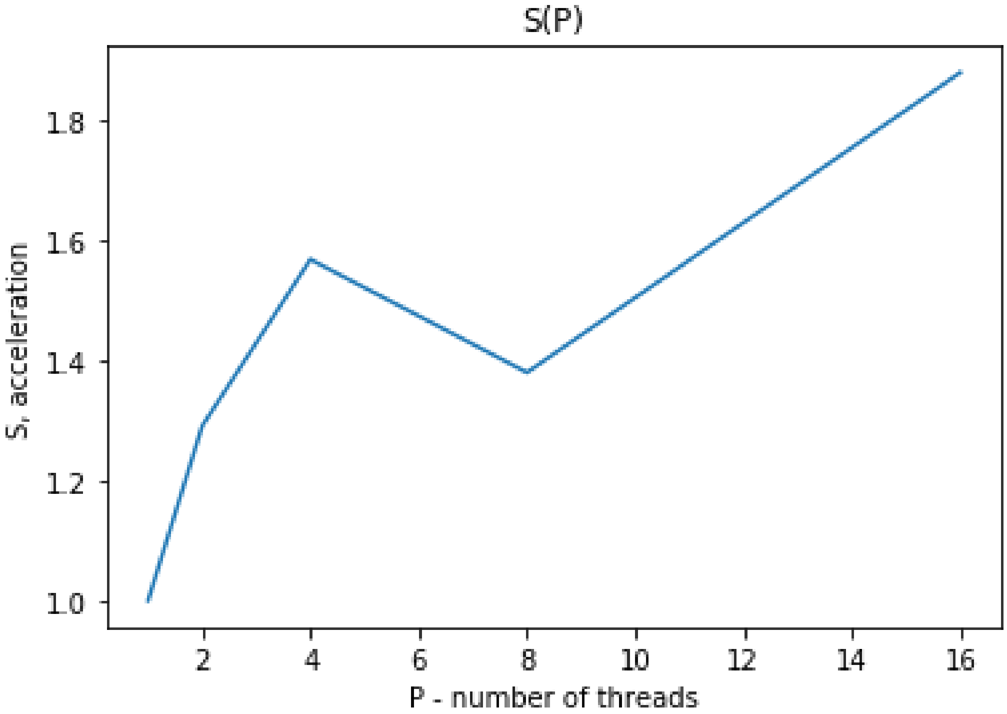
```
In [2]: # N = 1000
        p = np.array([1, 2, 4, 8, 16])
        mean_time = []

        # time values with P = 1
        data = np.array([0.024618, 0.042570, 0.049720, 0.037575, 0.034624,
                         0.035595, 0.074592, 0.043650, 0.037816, 0.083169])
        t_1 = data.mean()
        mean_time.append(t_1)

        #time values with P = 2
        data = np.array([0.069016, 0.024386, 0.035974, 0.025372, 0.061552,
                         0.024518, 0.020170, 0.056259, 0.018882, 0.023225])
        mean_time.append(data.mean())

        #time values with P = 4
        data = np.array([0.048135, 0.018452, 0.024613, 0.016405, 0.031424,
                         0.017200, 0.035382, 0.041523, 0.038055, 0.024540])
        mean_time.append(data.mean())

        #time values with P = 8
        data = np.array([0.048447, 0.044716, 0.017841, 0.026257, 0.037932,
                         0.028252, 0.027284, 0.024254, 0.040250, 0.040903])
        mean_time.append(data.mean())

        #time values with P = 16
        data = np.array([0.032631, 0.019313, 0.023957, 0.046037, 0.019929,
                         0.017501, 0.017719, 0.023065, 0.024425, 0.022183])
        mean_time.append(data.mean())

        s = t_1 / mean_time
        e = s / p
```

N = const

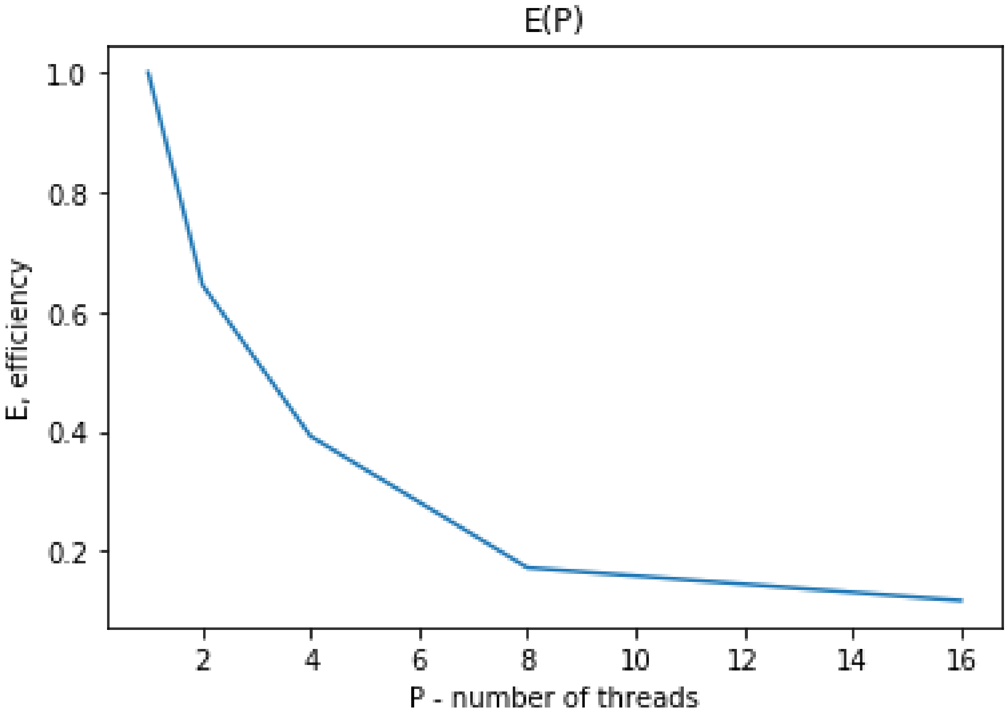
```
In [3]: plt.title("T(P)")
        plt.xlabel("P - number of threads")
        plt.ylabel("T, s - process time")
        plt.plot(p, mean_time)
        plt.show()
```



```
In [4]: plt.title("S(P)")
        plt.xlabel("P - number of threads")
        plt.ylabel("S, acceleration")
        plt.plot(p, s)
        plt.show()
```



```
In [5]: plt.title("E(P)")
        plt.xlabel("P - number of threads")
        plt.ylabel("E, efficiency")
        plt.plot(p, e)
        plt.show()
```


E(P)
1.0
0.8
0.6
0.4
0.2
E, efficiency
2
4
6
8
10
12
14
16
P - number of threads