In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [2]:
```
stats = pd.read_csv('test1.txt',
                    names=['E', 'S', 'T', 'a', 'b', 'x', 'N', 'p', 'P'
                    sep = " ")
stats['T'] = list(map((lambda x: float(x[:-1])), stats['T']))
stats.head()
```

Out[2]:

	E	S	T	a	b	x	N	p	P
0	49.900	2541.148000	0.061793	0	100	50	1000	0.5	1
1	50.400	2494.815000	0.109222	0	100	50	2000	0.5	1
2	50.000	2484.498667	0.158519	0	100	50	3000	0.5	1
3	49.475	2497.104500	0.220327	0	100	50	4000	0.5	1
4	49.260	2493.411600	0.281384	0	100	50	5000	0.5	1

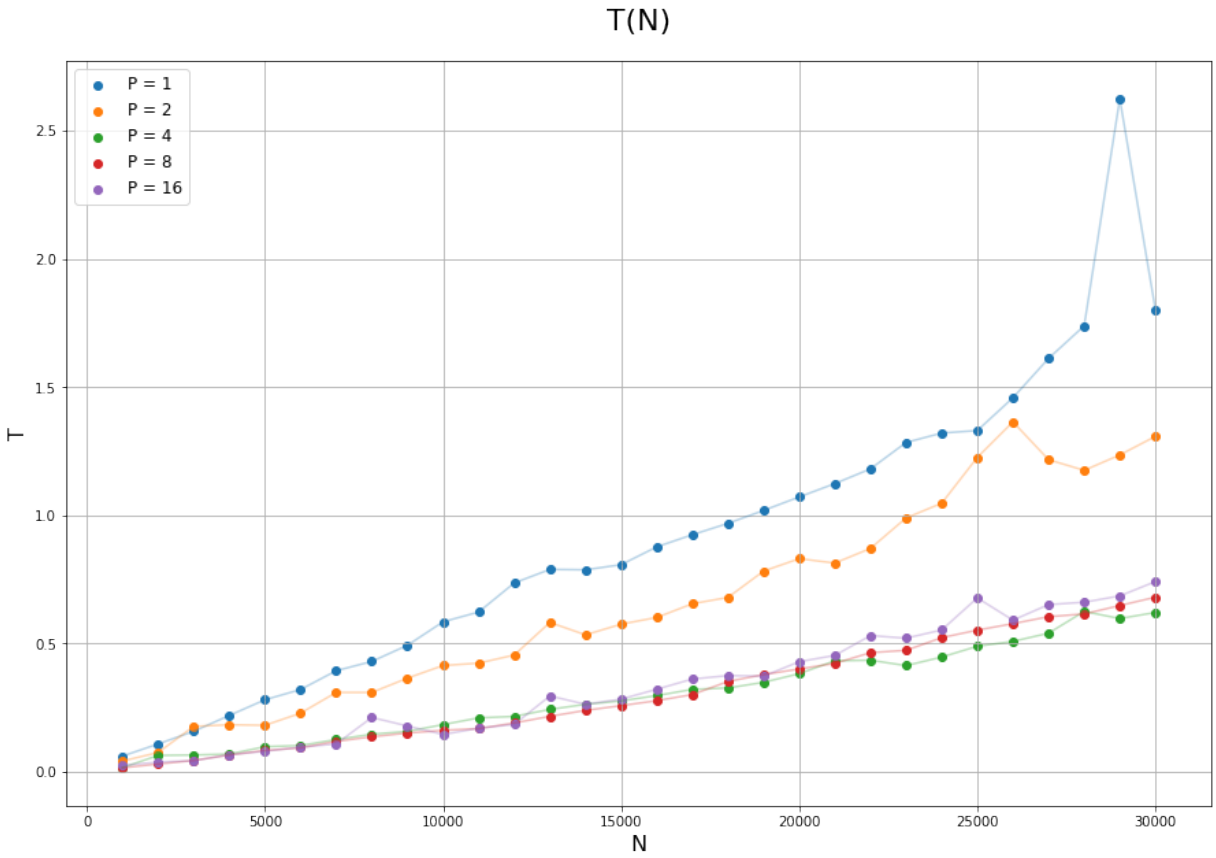
In [3]:
```
data = []
for i in [1, 2, 4, 8, 16]:
    data.append((i, stats[stats['P'] == i]))
```

In [4]:
```
plt.figure(figsize=(15, 10))

for data_i in data:
    plt.plot(data_i[1]['N'], np.array(data_i[1]['T']), alpha=0.3, label
    plt.scatter(data_i[1]['N'], np.array(data_i[1]['T']), label='P = '

plt.title('T(N)', y=1.03, fontsize=22)
plt.xlabel('N', fontsize=16)
plt.ylabel('T', fontsize=16)

plt.legend(fontsize=12)
plt.grid()
plt.show()
```

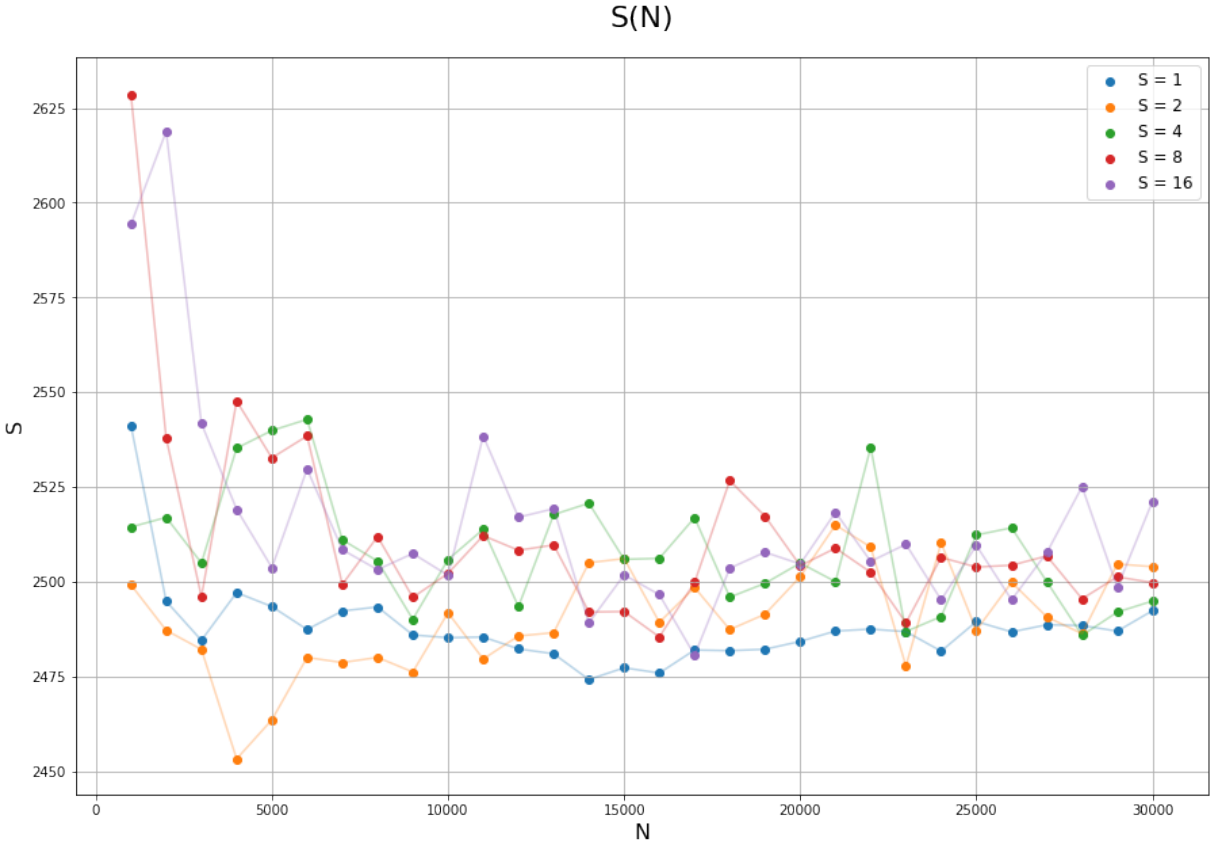


In [5]:
```
plt.figure(figsize=(15, 10))

for data_i in data:
    plt.plot(data_i[1]['N'], np.array(data_i[1]['S']), alpha=0.3, label
    plt.scatter(data_i[1]['N'], np.array(data_i[1]['S']), label='S = '

plt.title('S(N)', y=1.03, fontsize=22)
plt.xlabel('N', fontsize=16)
plt.ylabel('S', fontsize=16)

plt.legend(fontsize=12)
plt.grid()
plt.show()
```

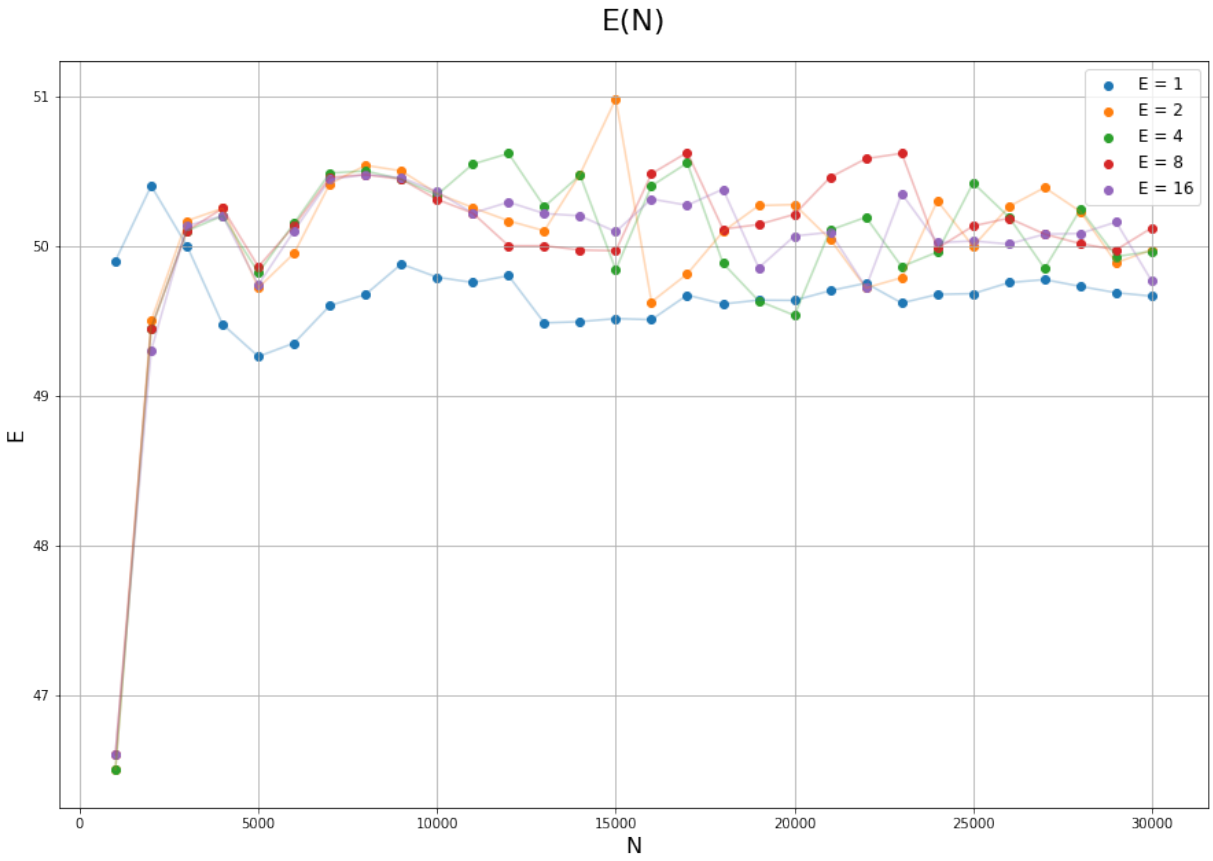


In [6]:
```
plt.figure(figsize=(15, 10))

for data_i in data:
    plt.plot(data_i[1]['N'], np.array(data_i[1]['E']), alpha=0.3, label
    plt.scatter(data_i[1]['N'], np.array(data_i[1]['E']), label='E = '

plt.title('E(N)', y=1.03, fontsize=22)
plt.xlabel('N', fontsize=16)
plt.ylabel('E', fontsize=16)

plt.legend(fontsize=12)
plt.grid()
plt.show()
```



In [7]:
```
stats = pd.read_csv('test2.txt',
                    names=['E', 'S', 'T', 'a', 'b', 'x', 'N', 'p', 'P'
                    sep = " ")
stats['T'] = list(map((lambda x: float(x[:-1])), stats['T']))
stats.head()
```

Out[7]:

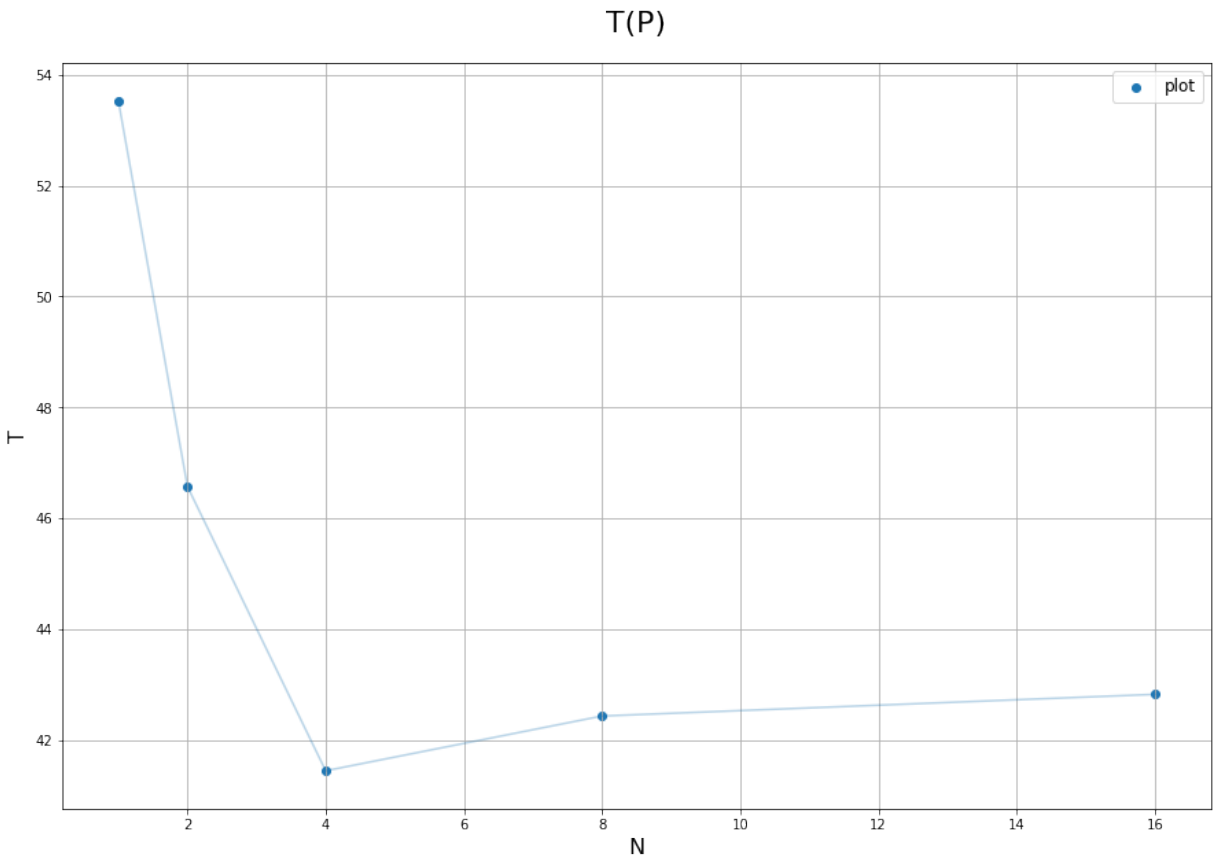
	E	S	T	a	b	x	N	p	P
0	49.9882	2497.977538	53.530575	0	100	50	1000000	0.5	1
1	50.0172	2500.135668	46.569041	0	100	50	1000000	0.5	2
2	49.9993	2500.257414	41.441273	0	100	50	1000000	0.5	4
3	50.0127	2498.056276	42.429647	0	100	50	1000000	0.5	8
4	50.0081	2499.166736	42.820681	0	100	50	1000000	0.5	16

In [8]:
```
plt.figure(figsize=(15, 10))

plt.plot(stats['P'], np.array(stats['T']), alpha=0.3, label=None)
plt.scatter(stats['P'], np.array(stats['T']), label='plot')

plt.title('T(P)', y=1.03, fontsize=22)
plt.xlabel('N', fontsize=16)
plt.ylabel('T', fontsize=16)

plt.legend(fontsize=12)
plt.grid()
plt.show()
```

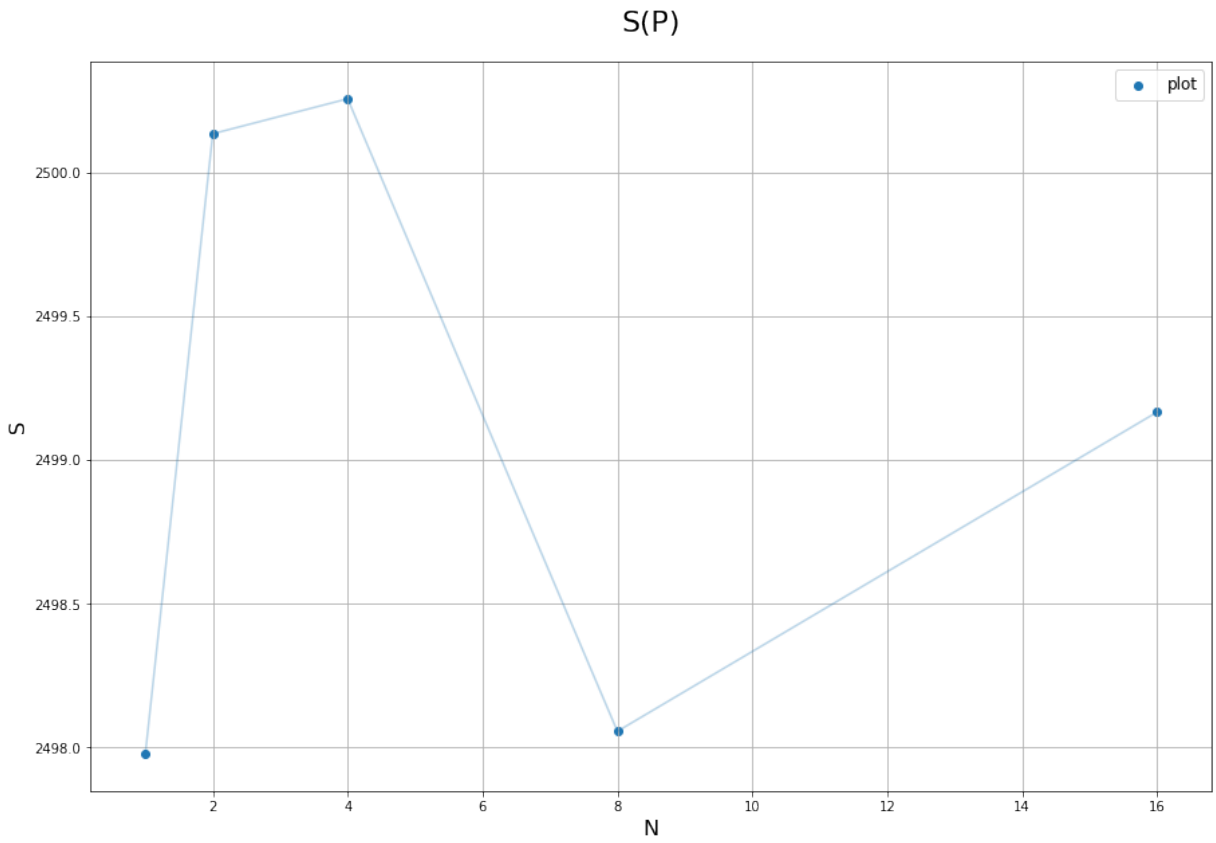


In [9]:
```
plt.figure(figsize=(15, 10))

plt.plot(stats['P'], np.array(stats['S']), alpha=0.3, label=None)
plt.scatter(stats['P'], np.array(stats['S']), label='plot')

plt.title('S(P)', y=1.03, fontsize=22)
plt.xlabel('N', fontsize=16)
plt.ylabel('S', fontsize=16)

plt.legend(fontsize=12)
plt.grid()
plt.show()
```



In [10]:
```
plt.figure(figsize=(15, 10))

plt.plot(stats['P'], np.array(stats['E']), alpha=0.3, label=None)
plt.scatter(stats['P'], np.array(stats['E']), label='plot')

plt.title('E(P)', y=1.03, fontsize=22)
plt.xlabel('N', fontsize=16)
plt.ylabel('E', fontsize=16)

plt.legend(fontsize=12)
plt.grid()
plt.show()
```