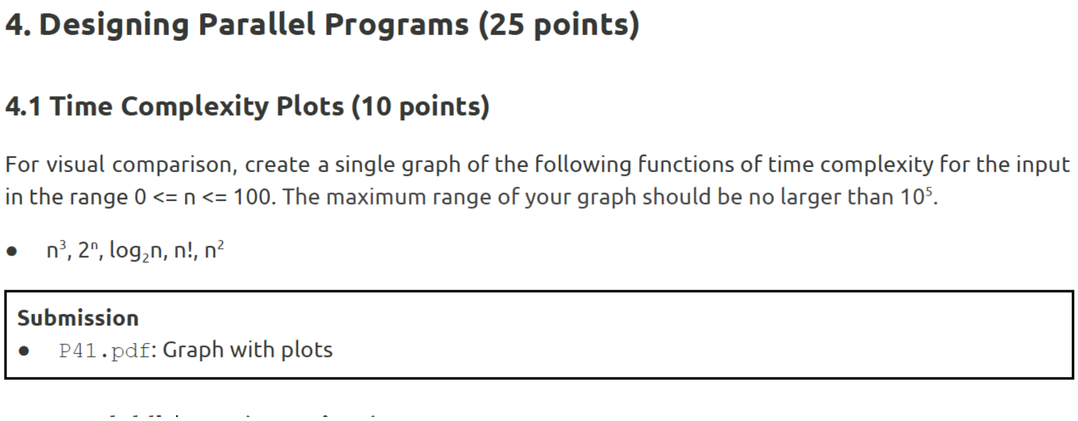


4. Designing Parallel Programs (25 points)

4.1 Time Complexity Plots (10 points)

For visual comparison, create a single graph of the following functions of time complexity for the input in the range 0 <= n <= 100. The maximum range of your graph should be no larger than 10^5.

- n^3, 2^n, $\log_2 n$, n!, n^2

Submission

- `P41.pdf`: Graph with plots

By Jiahui Tang

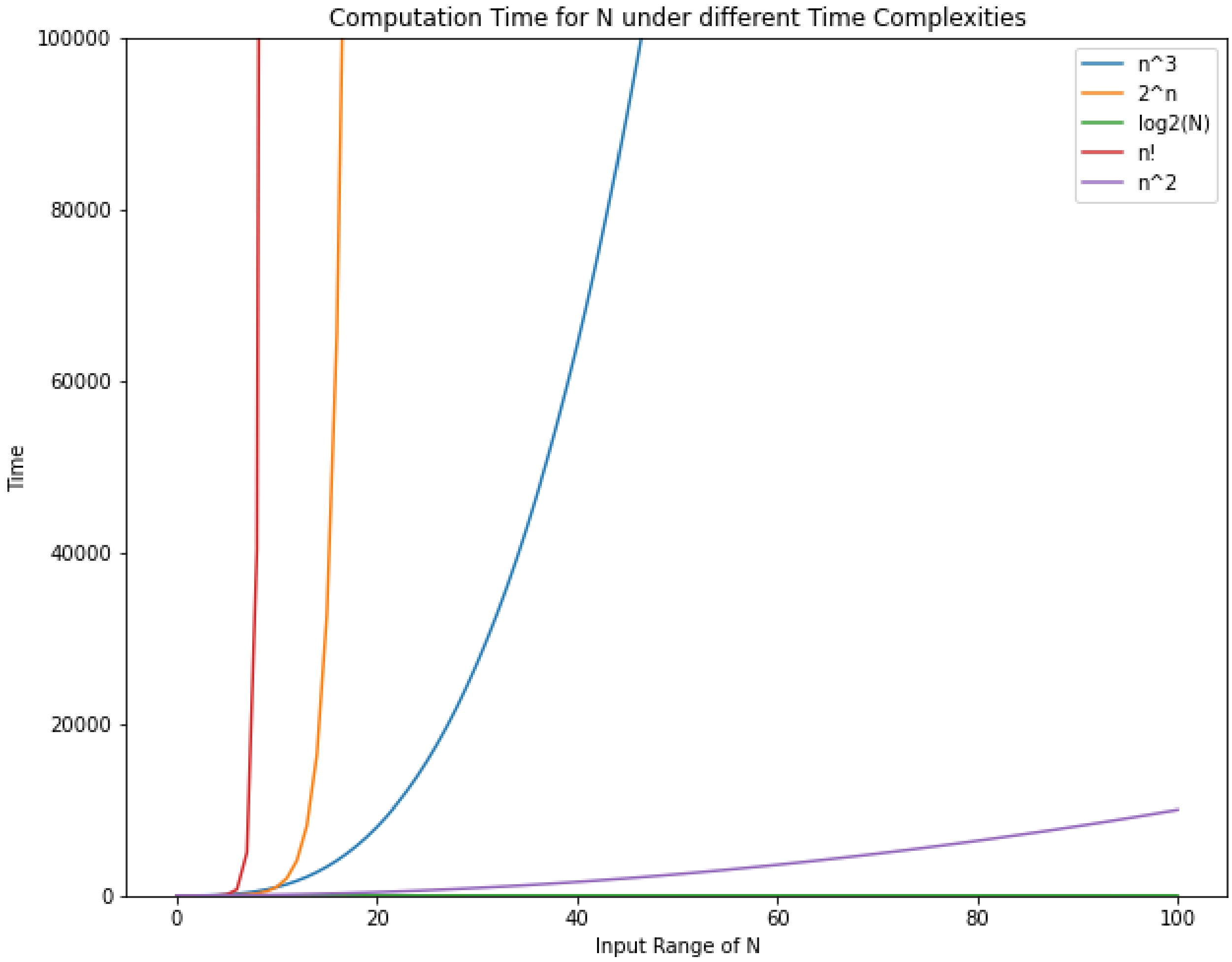
In [20]:

```python
import matplotlib.pyplot as plt
import numpy as np
import math

n = np.linspace(0, 100, 101)
n_pow_3 = np.power(n,3)
two_exp = np.power(2,n)
log2_n = np.log2(n)
n_fatorial = [np.math.factorial(i) for i in n]
n_pow_2 = np.power(n,2)

# plot
plt.figure(figsize = (10,8))
#plt.plot(n, n, label="Linear O(n)")

plt.plot(n, n_pow_3, label="n^3")
plt.plot(n, two_exp_lst, label="2^n")
plt.plot(n, log2_n, label="log2(N)")
plt.plot(n, n_fatorial, label="n!")
plt.plot(n, n_pow_2, label="n^2")
plt.ylim(0, 100000)


plt.xlabel('Input Range of N')
plt.ylabel('Time')
plt.title("Computation Time for N under different Time Complexities")
plt.legend()
```

```
<ipython-input-20-0df5fd5d7e3d>:8: RuntimeWarning: divide by zero encountered in
log2
  log2_n = np.log2(n)
```

Out[20]:

```
<matplotlib.legend.Legend at 0x7f98307fc910>
```


Computation Time for N under different Time Complexities
Time
100000
80000
60000
40000
20000
0
0
20
40
60
80
100
Input Range of N
n^3
2^n
log2(N)
n!
n^2

In [ ]: