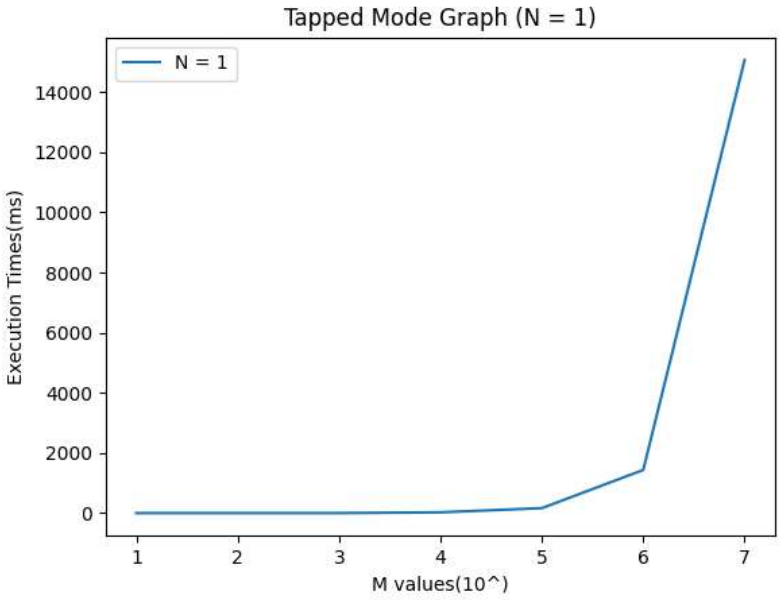
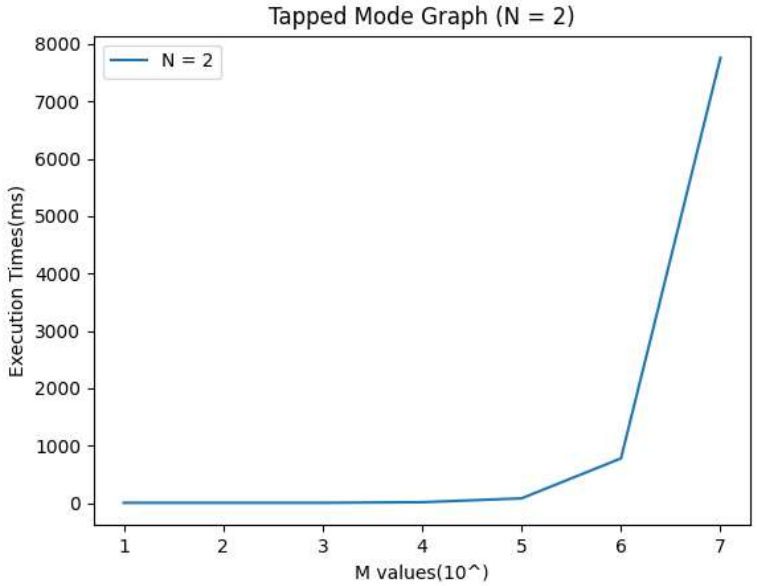
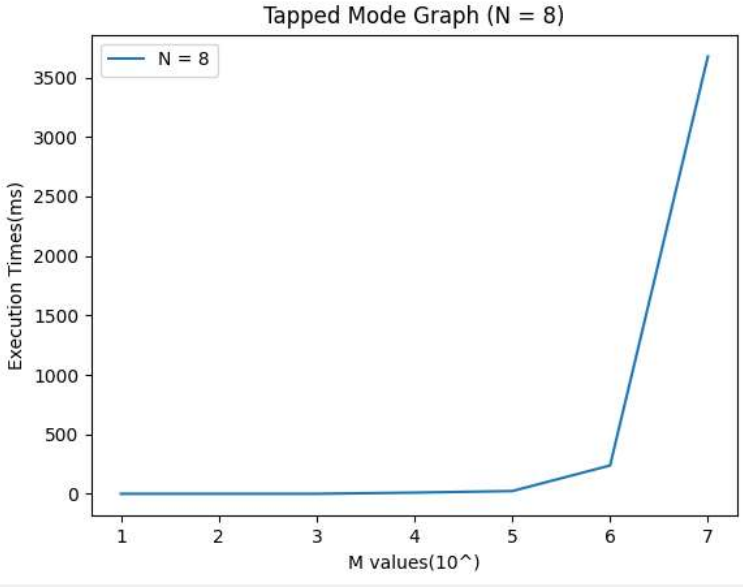
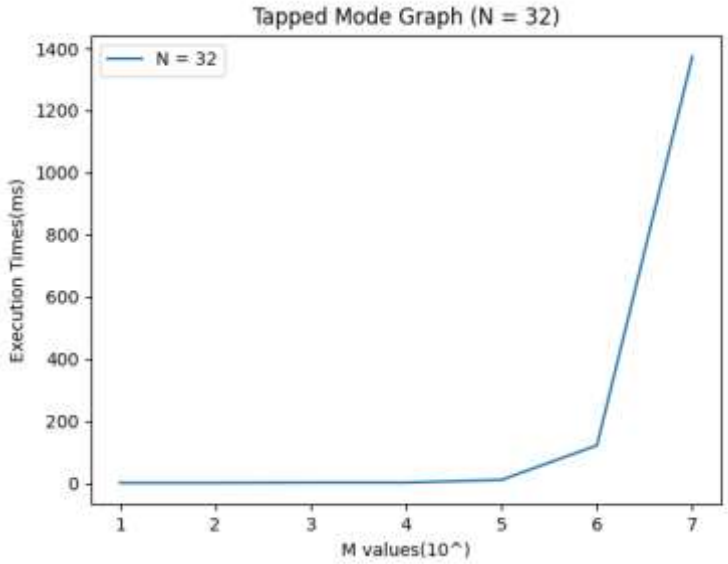
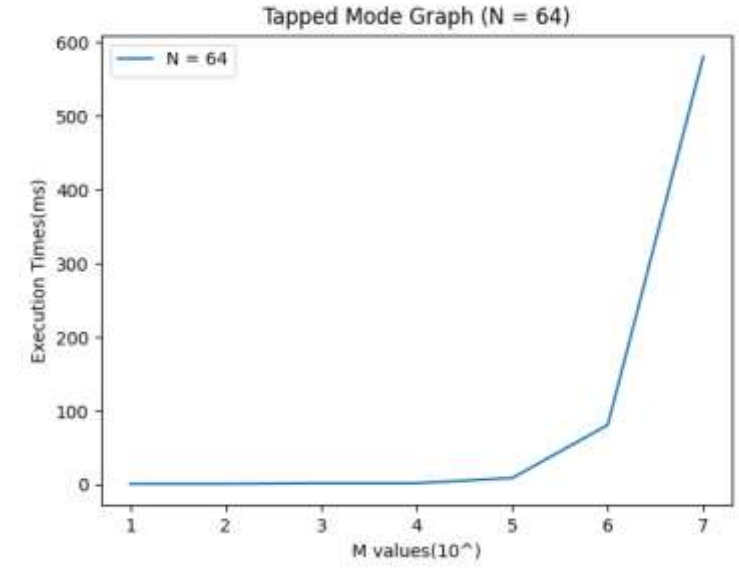
Muhammed Emre YILDIZ
21702825
Project-01

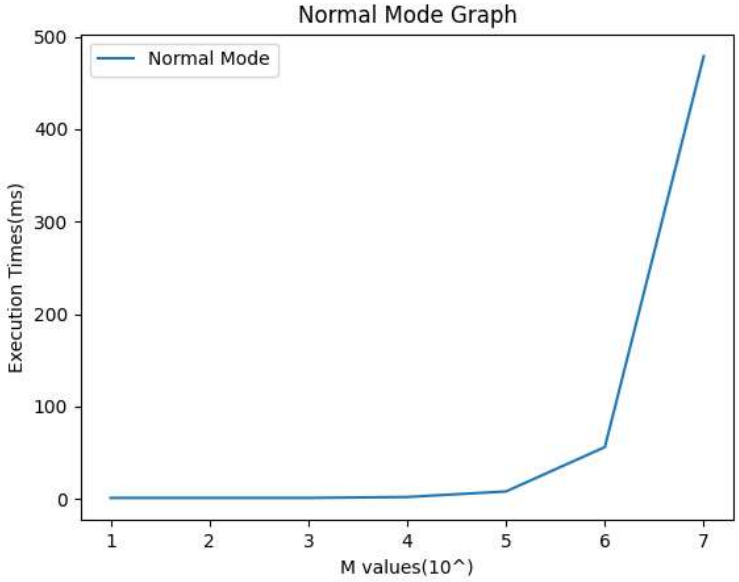

Tapped Mode Graph (N = 1)
N = 1
Execution Times(ms)
0
2000
4000
6000
8000
10000
12000
14000
1
2
3
4
5
6
7
M values(10^)

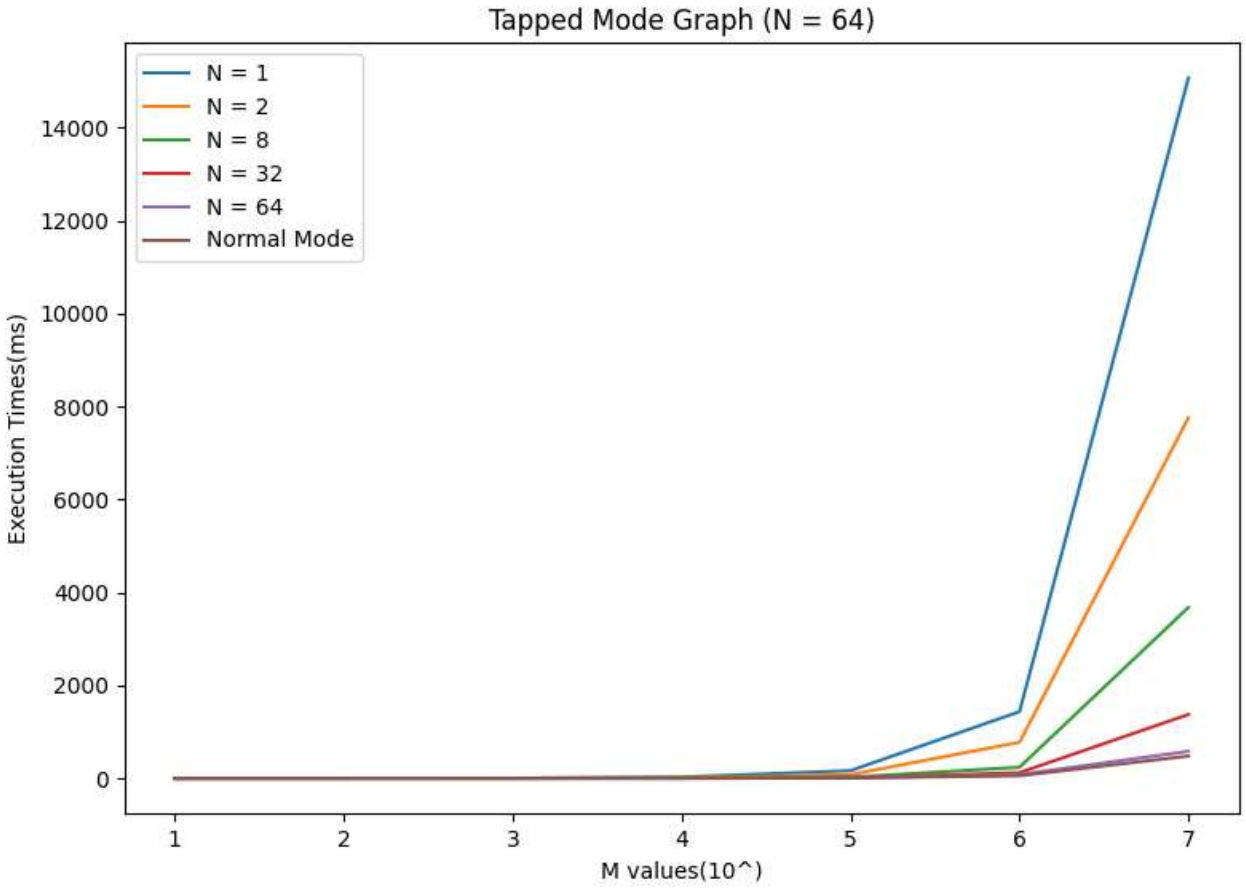

Tapped Mode Graph (N = 2)
N = 2
Execution Times(ms)
0
1000
2000
3000
4000
5000
6000
7000
8000
1
2
3
4
5
6
7
M values(10^)


Tapped Mode Graph (N = 8)
N = 8
Execution Times(ms)
0
500
1000
1500
2000
2500
3000
3500
1
2
3
4
5
6
7
M values(10^)


Tapped Mode Graph (N = 32)
N = 32
Execution Times(ms)
M values(10^)
0
200
400
600
800
1000
1200
1400
1
2
3
4
5
6
7


Tapped Mode Graph (N = 64)
N = 64
Execution Times(ms)
M values(10^)
0
100
200
300
400
500
600
1
2
3
4
5
6
7


Normal Mode Graph
Normal Mode
Execution Times(ms)
M values(10^)
0
100
200
300
400
500
1
2
3
4
5
6
7




## Conclusion:

Firstly, N =1 is runned means parent reads 1 byte from child1 and writes 1 byto to child 2 with 2 pipes. After that N = 2, N = 8, N = 32 and N = 64 are used because of the see differences between graphs. When N value increases, time consuming decreases because parent can write more bites respectively. Also, as we expected, when M value is increased the execution time increased exponentially. Because in order to creating random alphanumeric values and write and read them are became more difficult for processors. It takes nearly O(n) for creating and O(n) for read and write means O(n^2). Therefore our graphs seem exponential graphs. When normal mode is used, there is no read and write byte to byte and just 1 pipe is used means consuming less time. Thus, we can get best times when we use normal mode. If last graph is considered it can obviously said that when M value is increased, reading and writing need more time and we should increase the N value means how many bytes that parent uses when reading and writing.