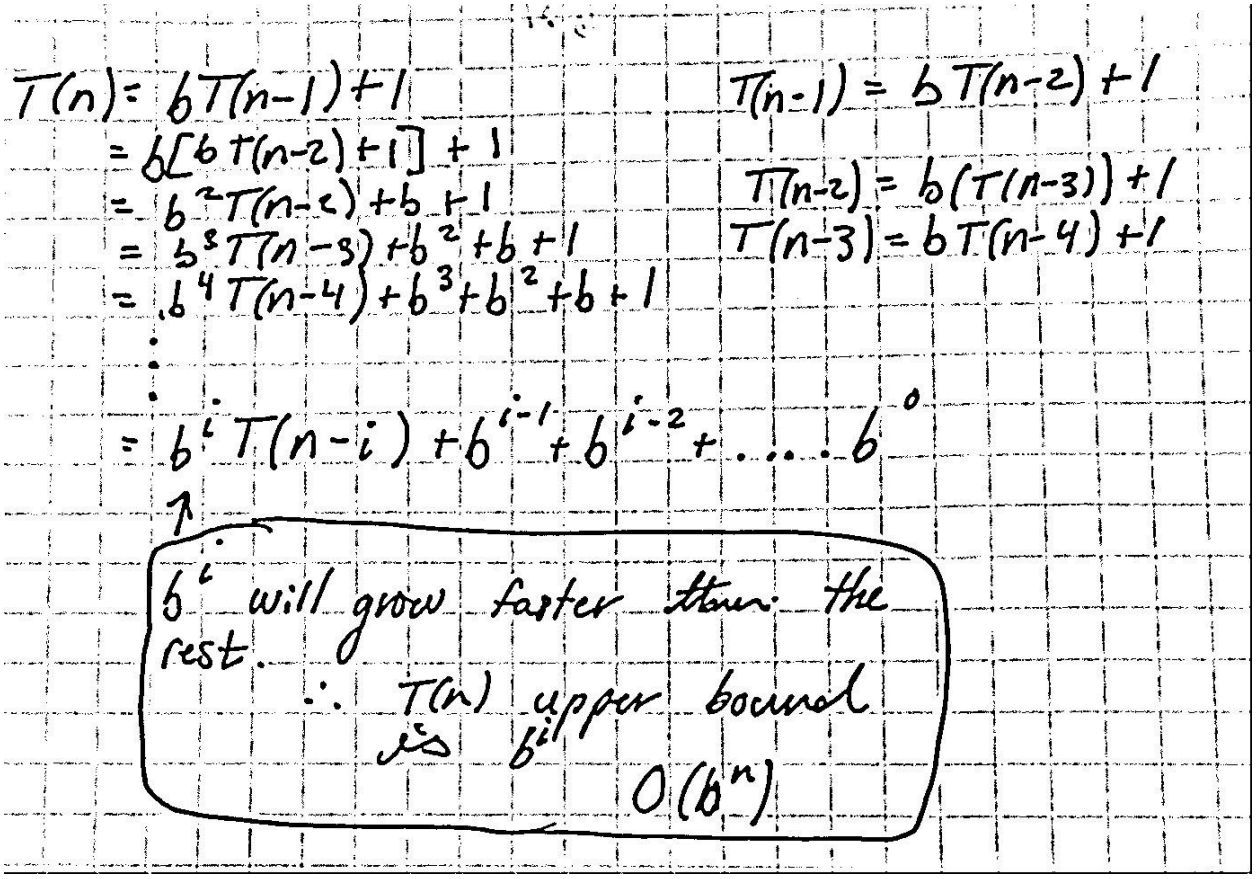
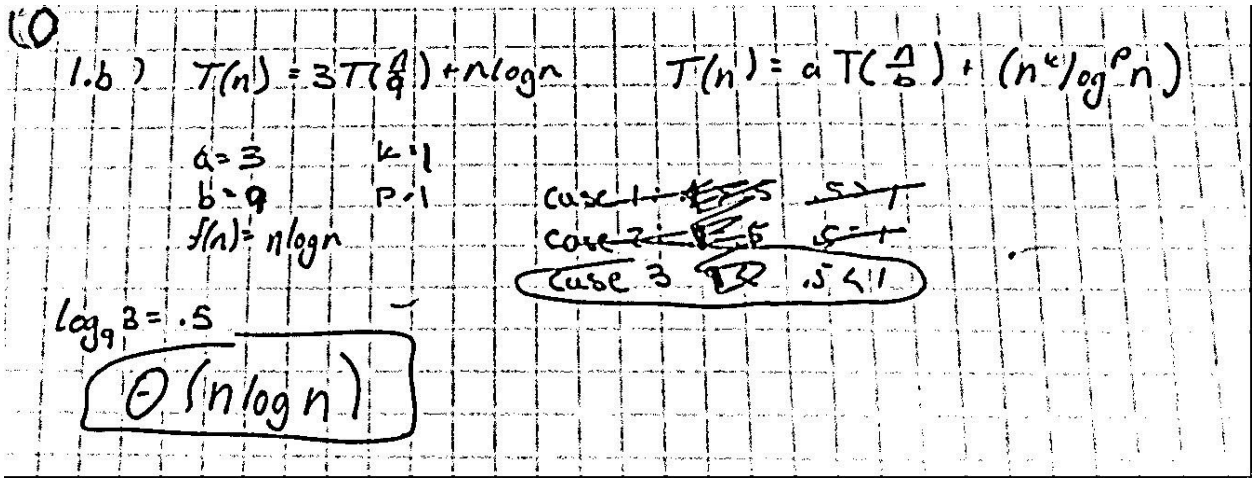
Harinder Gakhal
1/21/20

1.a

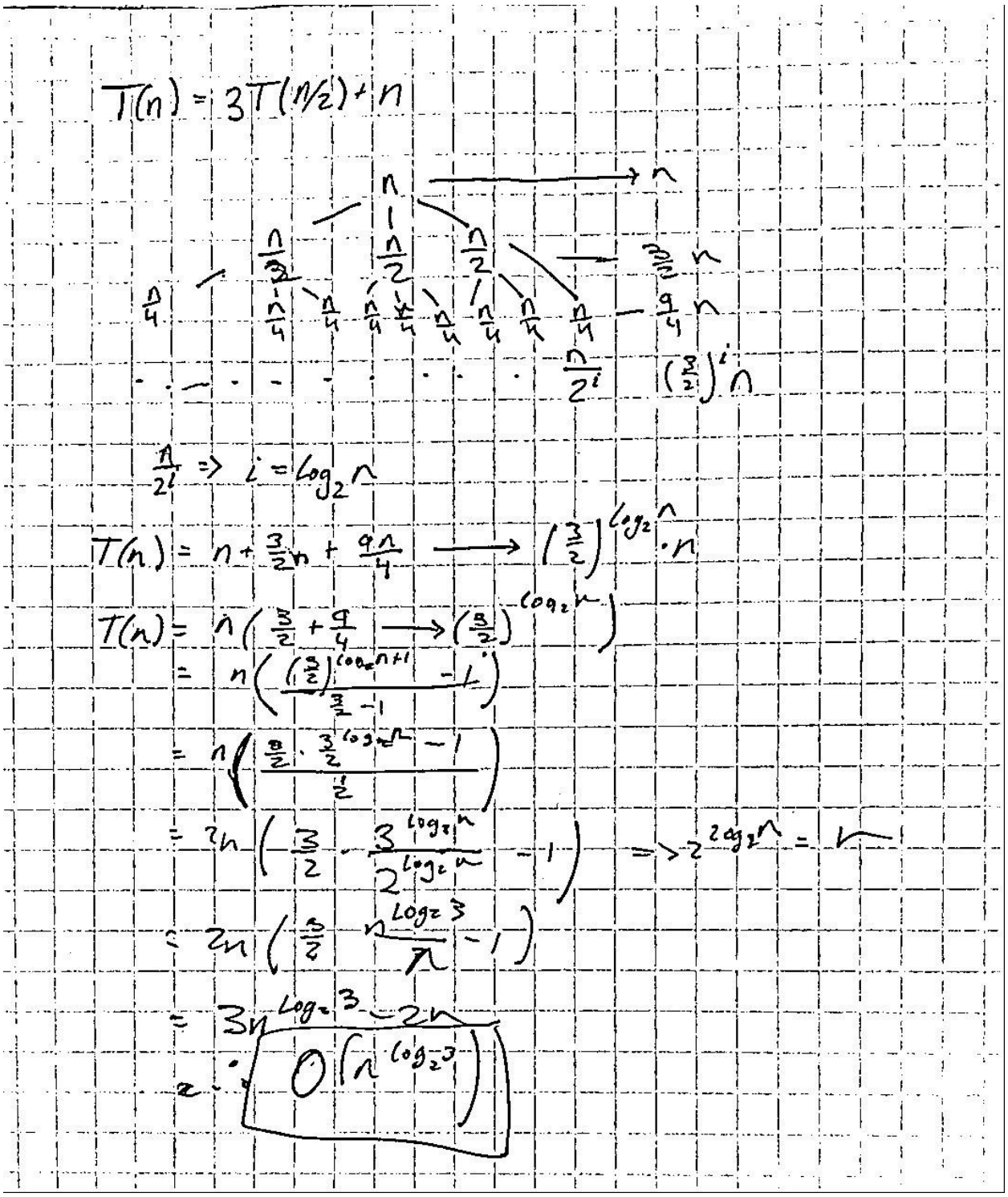


1b


1.b) $T(n) = 3T(\frac{n}{9}) + n\log n$
$T(n) = aT(\frac{n}{b}) + (n^k \log^p n)$
a = 3
b = 9
f(n) = nlogn
k = 1
p = 1
Case 1: ~~$f(n) = O(n^{\log_b a - \varepsilon})$~~ ~~$\log_b a > 1$~~
Case 2: ~~$f(n) = \Theta(n^{\log_b a} \log^k n)$~~ ~~$k = 1$~~
Case 3 $f(n) = \Omega(n^{\log_b a + \varepsilon})$ $.5 < 1$
$\log_9 3 = .5$
$\Theta(n \log n)$

```
MaxSum(A[0..n])

    max ← A[0]
    maxEnd ← A[0]

    for(int i=0; i< A.length; i++)

        if(maxEnd+A[i] > A[i])
            maxEnd += A[i]
        else
            maxEnd = A[i]

        if(max <= maxEnd)
            max = maxEnd

    return max
```

3a


$T(n) = 3T(n/2) + n$
$\frac{n}{2^i} \Rightarrow i = \log_2 n$
$T(n) = n + \frac{3}{2}n + \frac{9n}{4} \longrightarrow \left(\frac{3}{2}\right)^{\log_2 n} \cdot n$
$T(n) = n\left(\frac{3}{2} + \frac{9}{4} \longrightarrow \left(\frac{3}{2}\right)^{\log_2 n}\right)$
$= n\left(\frac{\left(\frac{3}{2}\right)^{\log_2 n + 1} - 1}{\frac{3}{2} - 1}\right)$
$= n\left(\frac{\frac{3}{2} \cdot \frac{3}{2}^{\log_2 n} - 1}{\frac{1}{2}}\right)$
$= 2n\left(\frac{3}{2} \cdot \frac{3^{\log_2 n}}{2^{\log_2 n}} - 1\right) \Rightarrow 2^{2\log_2 n} = n$
$= 2n\left(\frac{3}{2} \cdot \frac{n^{\log_2 3}}{n} - 1\right)$
$= 3n^{\log_2 3} - 2n$
$= O(n^{\log_2 3})$

3b

(b) $T(n) = 3T(\frac{n}{2}) + n$

base case $n=0$

$T(0) = (0)^{\log_2 3}$

$T(0) = 0$

for $n = 2^i$ we can assume it is true

for $n = 2^i$

Inductive step

$n = 2^{i+1}$

$T(2^{i+1}) = 3T(\frac{2^{i+1}}{2}) + 2^{i+1}$

$= 3T(\frac{2i+2}{2}) + 2^{i+1}$

$= 3T(2^i) + 2^{i+1}$

$= 3(2^i)^{\log_2 3} + 2^{i+1}$

$= (2^i \cdot 2)^{\log_2 3}$

$= (2^{i+1})^{\log_2 3}$

$T(2^{i+1})$

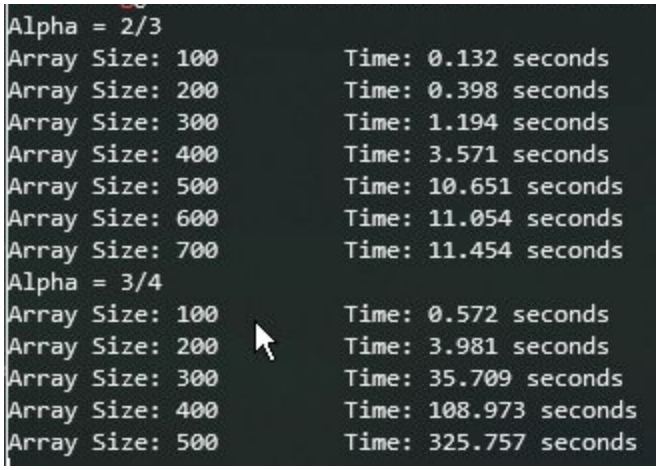
∴ it is true for 2^{i+1}

4b and 4c

4a. badSort will fail for alpha $\leq \frac{1}{2}$ because the algorithem will not be able to shuffle propely.

4b - We had to change the rounding to be round down instead of up. The algorithem did not finish and ran into error.

5b



5c

Experimental:

Trial	Alpha	Number of Values	BadSort Time
1	2/3	100	0.132
2	2/3	200	0.398
3	2/3	300	1.194
4	2/3	400	3.571
5	2/3	500	10.651

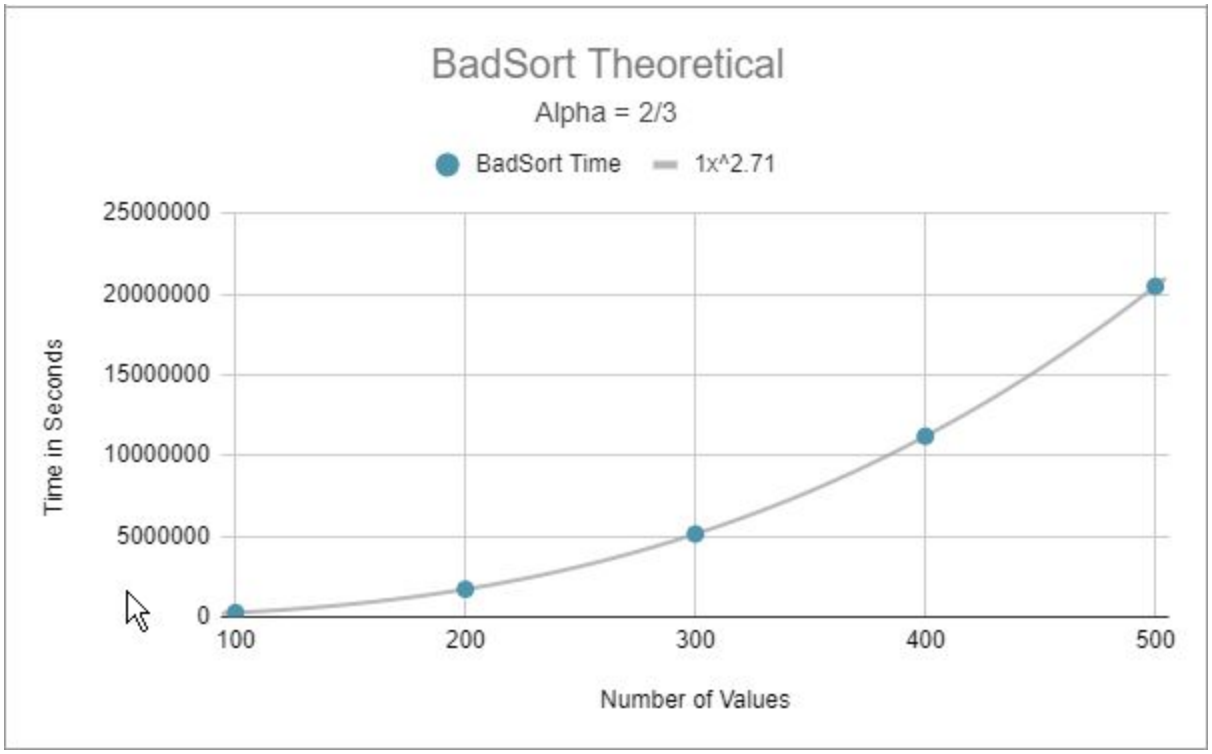
Trial	Alpha	Number of Values	BadSort Time
1	3/4	100	0.572
2	3/4	200	3.981
3	3/4	300	35.709
4	3/4	400	108.973
5	3/4	500	325.757

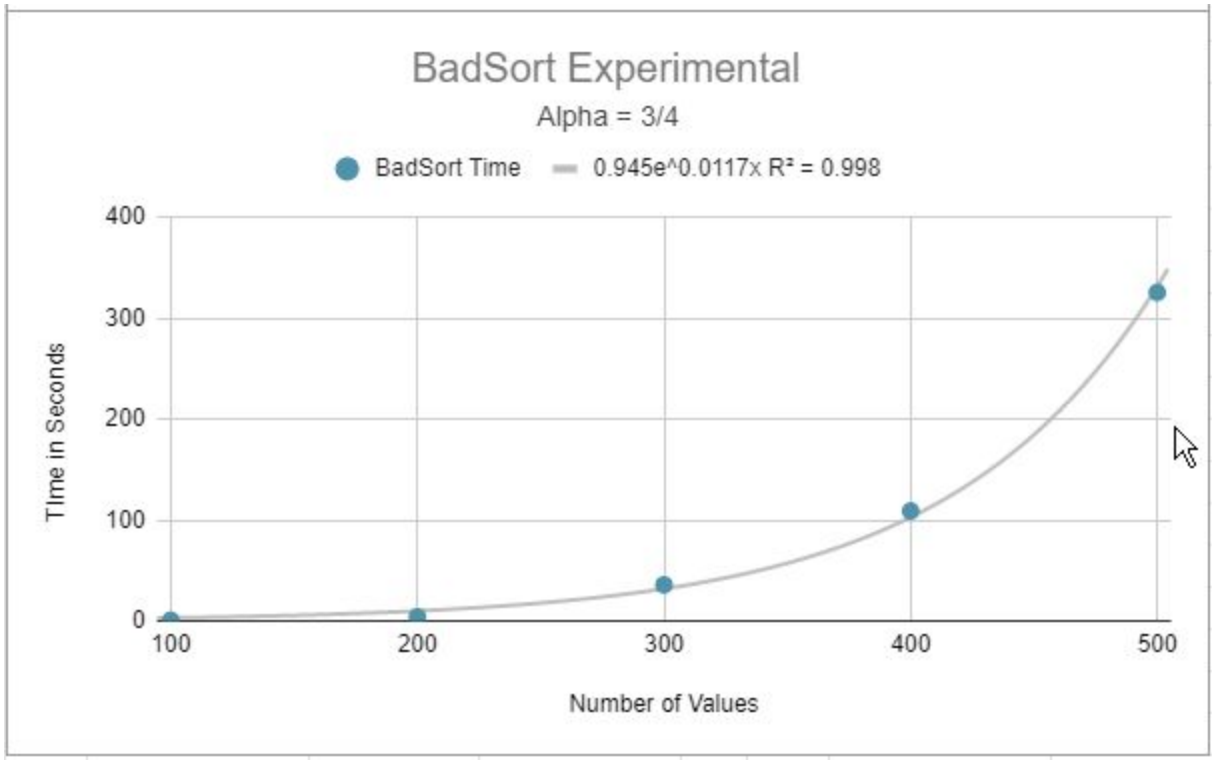
Theoretical:

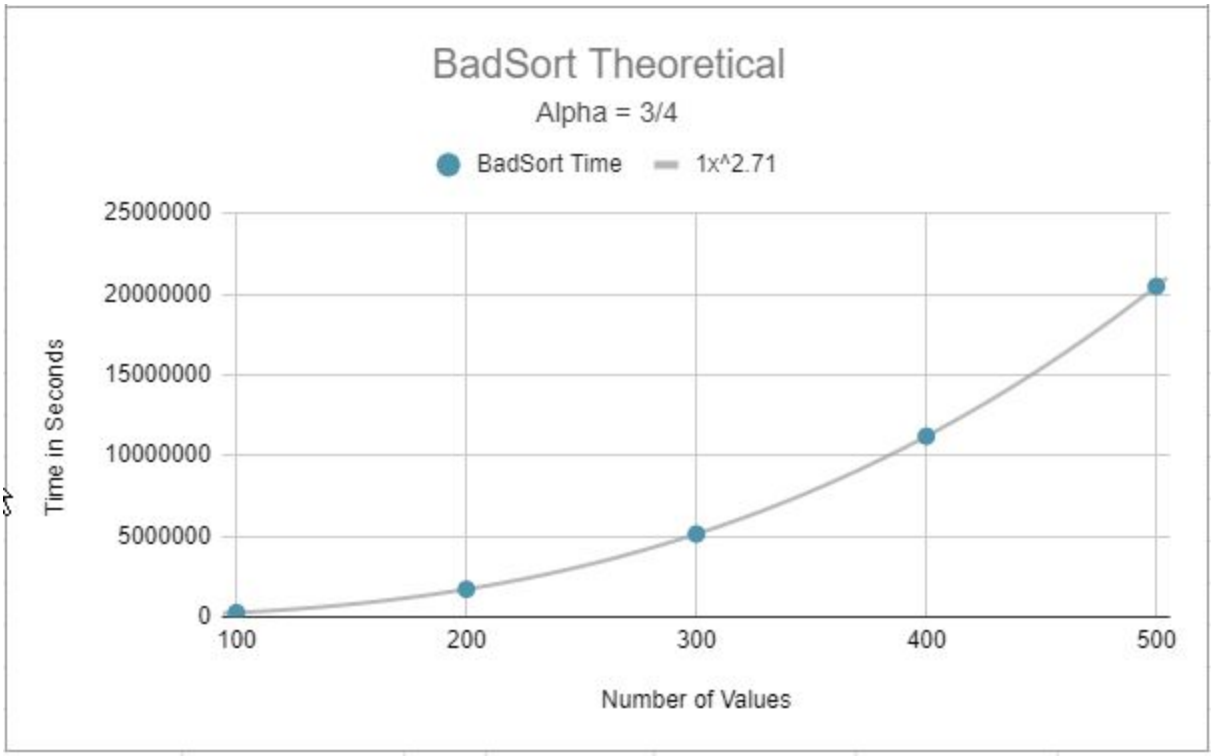
Number of Values	Alpha	BadSort Time
100	2/3	261818.3008
200	2/3	1711946.775
300	2/3	5134775.721
400	2/3	11193876.64
500	2/3	20488480.08

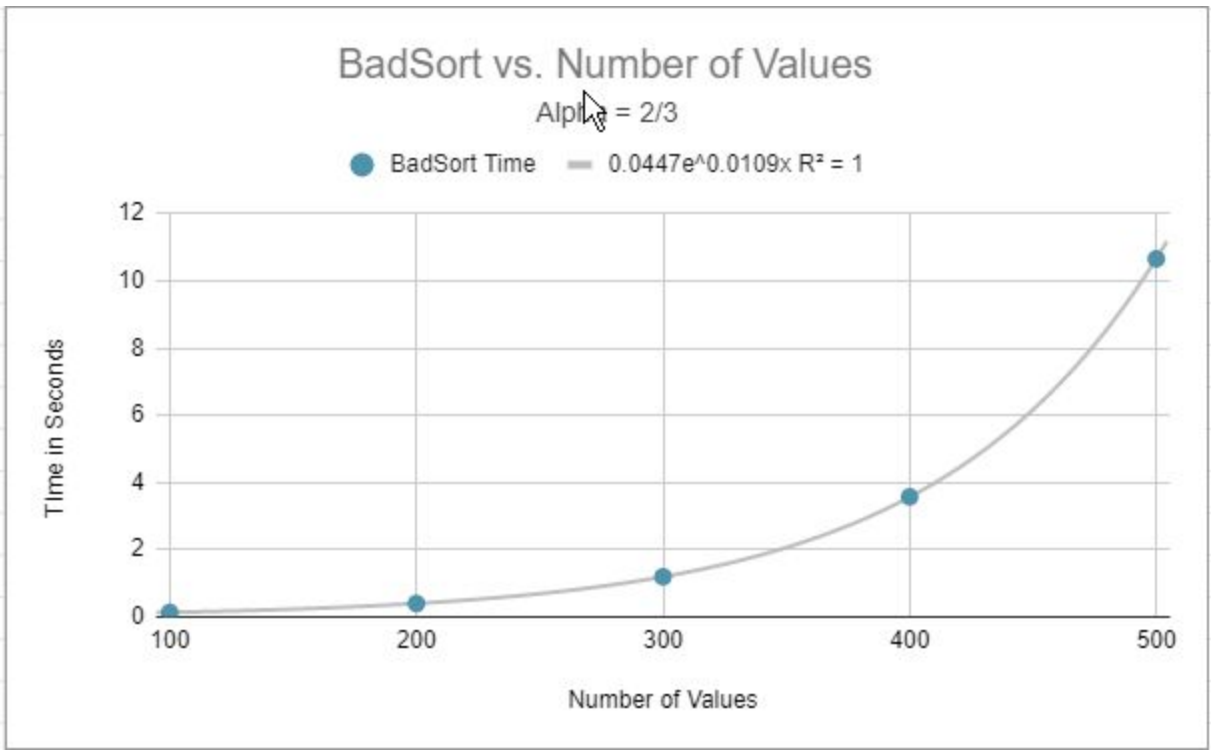
Number of Values	Alpha	BadSort Time
100	3/4	261818.3008
200	3/4	1711946.775
300	3/4	5134775.721
400	3/4	11193876.64
500	3/4	20488480.08

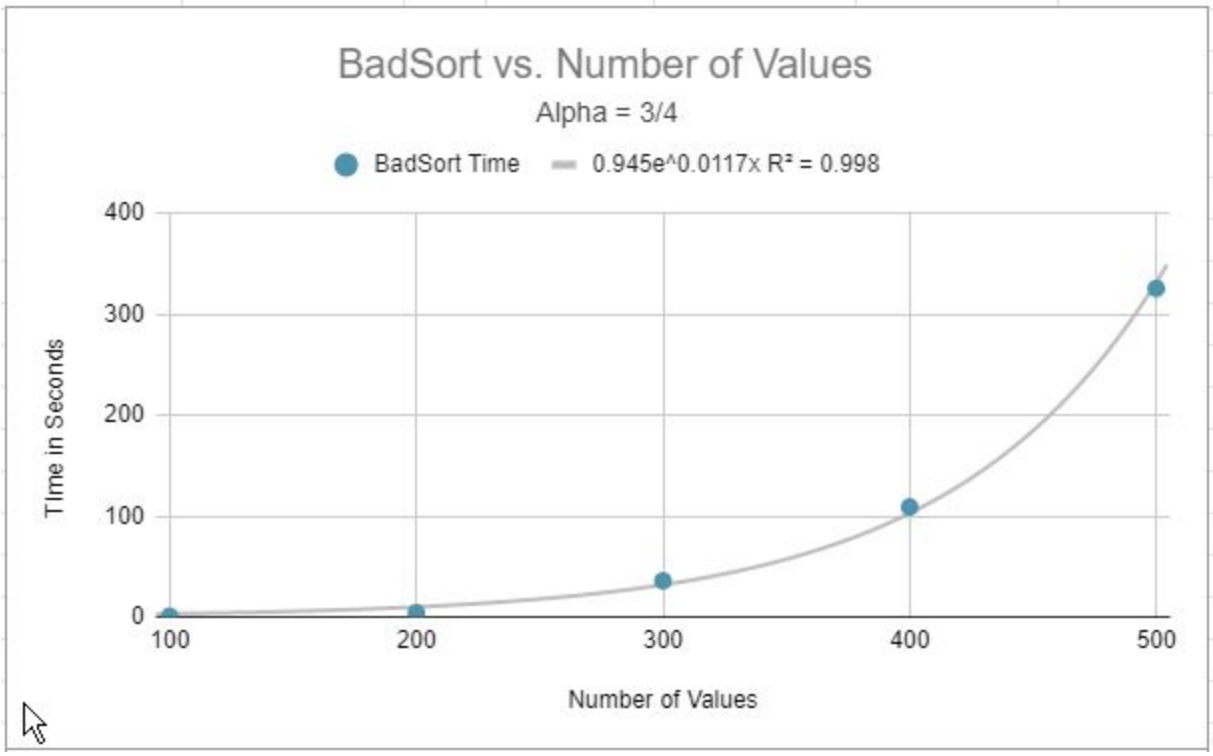
Graphs:

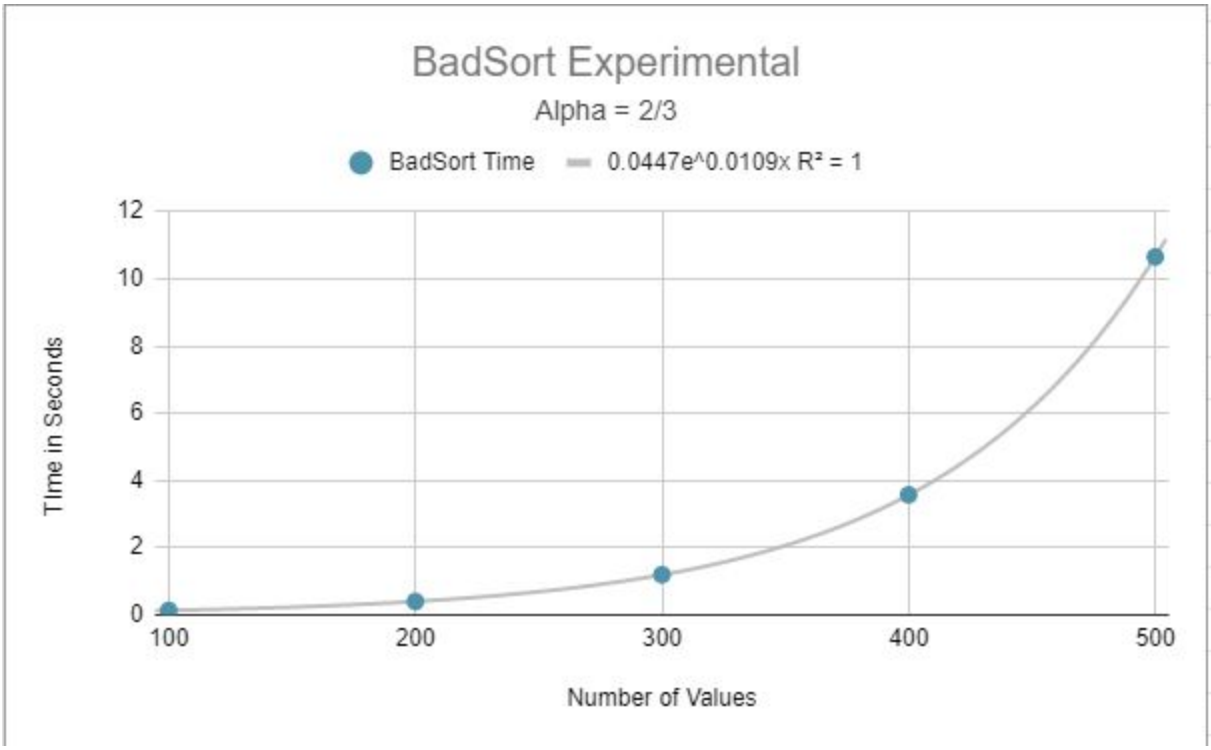

BadSort Theoretical
Alpha = 2/3
BadSort Time
1x^2.71
25000000
20000000
15000000
10000000
5000000
0
Time in Seconds
100
200
300
400
500
Number of Values


BadSort Experimental
Alpha = 3/4
BadSort Time
0.945e^0.0117x R² = 0.998
400
300
200
100
0
Time in Seconds
100
200
300
400
500
Number of Values


BadSort Theoretical
Alpha = 3/4
BadSort Time
1x^2.71
25000000
20000000
15000000
10000000
5000000
0
Time in Seconds
100
200
300
400
500
Number of Values


BadSort vs. Number of Values
Alpha = 2/3
BadSort Time
0.0447e^0.0109x R² = 1
12
10
8
6
4
2
0
TIme in Seconds
100
200
300
400
500
Number of Values





5d

Alpha = ⅔ is much faster than alpha = ¾ for large values.