Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Table of Contents

Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

THEORETICAL ANALYSIS

Basic operation is the comparison marked as (1)

Analyze B(n)

B(n) is for an array which has 0 for all its elements. Because, although comparison marked as (1) will be called exactly n times no matter the array the other basic operations will cause this case to run fastest. $\theta(f(n) \in \theta(n)$ if the basic operation is the comparison marked as (1).

Analyze W(n)

W(n) is for an array which has 2 for all its elements. Because, although comparison marked as (1) will be called exactly n times no matter the array the other basic operations will cause this case to run slowest. $\theta(f(n) \in \theta(n)$ if the basic operation is the comparison marked as (1).

Analyze A(n)

There is no such case that comparison marked as (1) doesn't called exactly n times. So no matter what the probability distribution is, average always will be called n times too. $(n + n + n)/3 = n$. So $\theta(f(n) \in \theta(n)$ if the basic operation is the comparison marked as (1).

Basic operations are the three assignments marked as (2)

Analyze B(n)

Best case is for an array which has 0 for all its elements. Because it has the lowest count of basic operations.

The while loop which the (2) marked operation is in of, will run exactly $\lceil \log_2(n+1) \rceil$ times. The upper loop is a for loop which will be called n times in the first iteration of the first for loop, then n-1, then n-2, decreasing by 1 until it reaches 0.

$$(n * (n+1) / 2) * \lceil \log_2(n+1) \rceil$$

This basic operation will be called exactly $(n^2 + 2n + 1)/2 * \lceil \log_2(n+1) \rceil$ times. $\theta(f(n) \in \theta(n^2 \log n)$ if the basic operation is the comparison marked as (2).

Analyze W(n)

W(n) is for an array which has 2 for all its elements. Because it has the lowest count of basic operations.

The inner for loop which the (2) marked operation is in, will be called 3 times in the first iteration of the first loop, then 8, then 15, until $(n+1)^2-1$. The most outer loop will iterate n times.

$$n * \sum_{1}^{n} ((n+1)^2 - 1) = n^4/3 + 3n^3/2 + 7n^2/6$$

This basic operation will be called exactly $n^4/3 + 3n^3/2 + 7n^2/6$ times. $\theta(f(n) \in \theta(n^4)$ if the basic operation is the comparison marked as (2).

Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Analyze A(n)

As shown in the B(n) and W(n) analyze, for the i = 0 case, the basic operation will be called $(n^2 + 2n + 1)/2 * \lceil \log_2(n+1) \rceil$ times. And for i = 2 case the basic operation will be called $n^4/3 + 3n^3/2 + 7n^2/6$ times.

For i = 1 case, the while loop which the (2) marked operation is in of, will run exactly $\lceil \log_2(n+1) \rceil$ times. Each of the three upper loops will be called exactly n times.

$n * n * n * \lceil \log_2(n+1) \rceil$

This basic operation will be called exactly $n^3 * \lceil \log_2(n+1) \rceil$ times. $\theta(f(n) \in \theta(n^3 * \log n)$ if the basic operation is the comparison marked as (2).

For the average case, since the probability distribution is uniform the average of these 3 cases should be equal to the A(n).

$A(n) = ((n * (n+1) / 2) * \lceil \log_2(n+1) \rceil + n * n * n * \lceil \log_2(n+1) \rceil + n^4/3 + 3n^3/2 + 7n^2/6)/3$

$A(n) = n^2 * \lceil \log_2(n+1) \rceil/6 + n * \lceil \log_2(n+1) \rceil/6 + n^3 * \lceil \log_2(n+1) \rceil/3 + n^4/9 + n^3/2 + 7n^2/18$

$\theta(f(n) \in \theta(n^4)$ if the basic operation is the comparison marked as (2).

Basic operation is two assignments marked as (3)

Analyze B(n)

The best case for this basic operation is an array which has 0 as all its elements. In that case the two assignments marked as (3) won't be called at all. B(n) = 0 if the basic operation is the comparison marked as (3).

Analyze W(n)

The worst case is for i = 2 and it will yield the exact same results as (2-W(n)) marked assignments. $W(n) = n^4/3 + 3n^3/2 + 7n^2/6$. $\theta(f(n) \in \theta(n^4)$ if the basic operation is the comparison marked as (3).

Analyze A(n)

In two out of three cases this basic operation will be called the same time as the (2) marked basic operations. Except for i = 0 in that case it won't be called at all. For i = 1, $n^4/3 + 3n^3/2 + 7n^2/6$ and for i = 2, $n * n * n * \lceil \log_2(n+1) \rceil$. For the average case, since the probability distribution is uniform the average of these 3 cases should be equal to the A(n).

$A(n) = (0 + n^4/3 + 3n^3/2 + 7n^2/6 + , n * n * n * \lceil \log_2(n+1) \rceil)/3$

$A(n) = n^4/9 + n^3/2 + n^3 * \lceil \log_2(n+1) \rceil/3 + 7n^2/18$

$\theta(f(n) \in \theta(n^4)$ if the basic operation is the comparison marked as (3).

Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Basic operations are the two loop incrementations marked as (4)

Analyze B(n)

The best case for this basic operation is an array which has 0 as all its elements. In that case the two assignments marked as (4) won't be called at all. B(n) = 0 if the basic operation is the comparison marked as (4).

Analyze W(n)

The worst case is for i = 1 for this basic operation. Each loop marked (4), the outer loop and the outmost loop iterates n times. $W(n) = n^3$. $\theta(f(n) \in \theta(n^3)$ if the basic operation is the comparison marked as (4).

Analyze A(n)

As shown in the B(n) and W(n) analyze, for the i = 0 case, the basic operation will be called 0 times. And for i = 1 case the basic operation will be called n^3 times.

For i = 2, both the loop marked (4) and the outer loop iterates n times. For i = 2 complexity is n^2. $\theta(f(n) \in \theta(n^2)$ if the basic operation is the comparison marked as (4).

For the average case, since the probability distribution is uniform the average of these 3 case should be equal to the A(n).

$A(n) = (0 + n^3 + n^2)/3$
$A(n) = n^3/3 + n^2/3$

$\theta(f(n) \in \theta(n^3)$ if the basic operation is the comparison marked as (4).

Basic operation is the assignment marked as (5)

Analyze B(n)

For the basic operation marked as (5), It doesn't called as long as the array doesn't include "0". So, B(n) = 0. But the absolute best case is for i = 1 for all elements. Because if i = 2 then 1 extra if check will get called.

Analyze W(n)

Worst case for this basic operation is for i = 0. As shown in the first marked operation. This operation will be called $(n^2 + n)/2$ times. $W(n) = (n^2 + n)/2$. $\theta(f(n) \in \theta(n^2)$ if the basic operation is the comparison marked as (5).

Analyze A(n)

For i = 0 complexity is $(n^2 + n)/2$. Both for i = 1 and i = 2 complexity is 0. So, the average is $(0 + 0 + (n^2 + n)/2)/3 = (n^2 + n)/6$. $A(n) = (n^2 + n)/6$. $\theta(f(n) \in \theta(n^2)$ if the basic operation is the comparison marked as (5).

IDENTIFICATION OF BASIC OPERATION(S)

The most suitable basic operation to choose is the assignment operations marked as (2). Because it gets called in every possible case and in every iteration there is.

Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

REAL EXECUTION

Best Case

N Size	Time Elapsed
1	0.00000 sec
5	0.00000 sec
10	0.00000 sec
25	0.00032 sec
50	0.00222 sec
75	0.00409 sec
100	0.00886 sec
150	0.02262 sec
200	0.03970 sec
250	0.06186 sec

Worst Case

N Size	Time Elapsed
1	0.00000 sec
5	0.00000 sec
10	0.00099 sec
25	0.01575 sec
50	0.26797 sec
75	1.30855 sec
100	4.22626 sec
150	21.52973 sec
200	76.90652 sec
250	209.67806 sec

Average Case

N Size	Time Elapsed
1	0.00000 sec
5	0.00000 sec
10	0.00057 sec
25	0.00710 sec
50	0.12868 sec
75	0.64771 sec
100	1.74158 sec
150	9.48504 sec
200	26.22953 sec
250	70.21483 sec

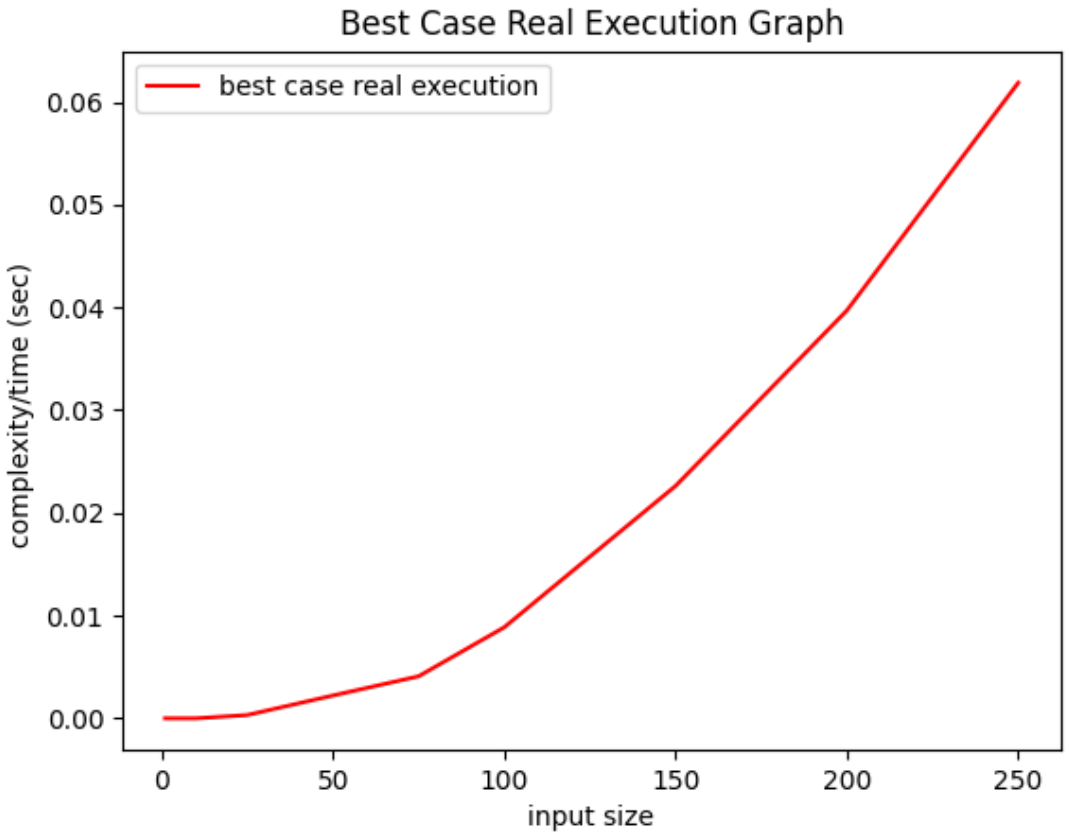
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

COMPARISON

Note: In the graphs : x-axis : input size , y-axis: time (second)

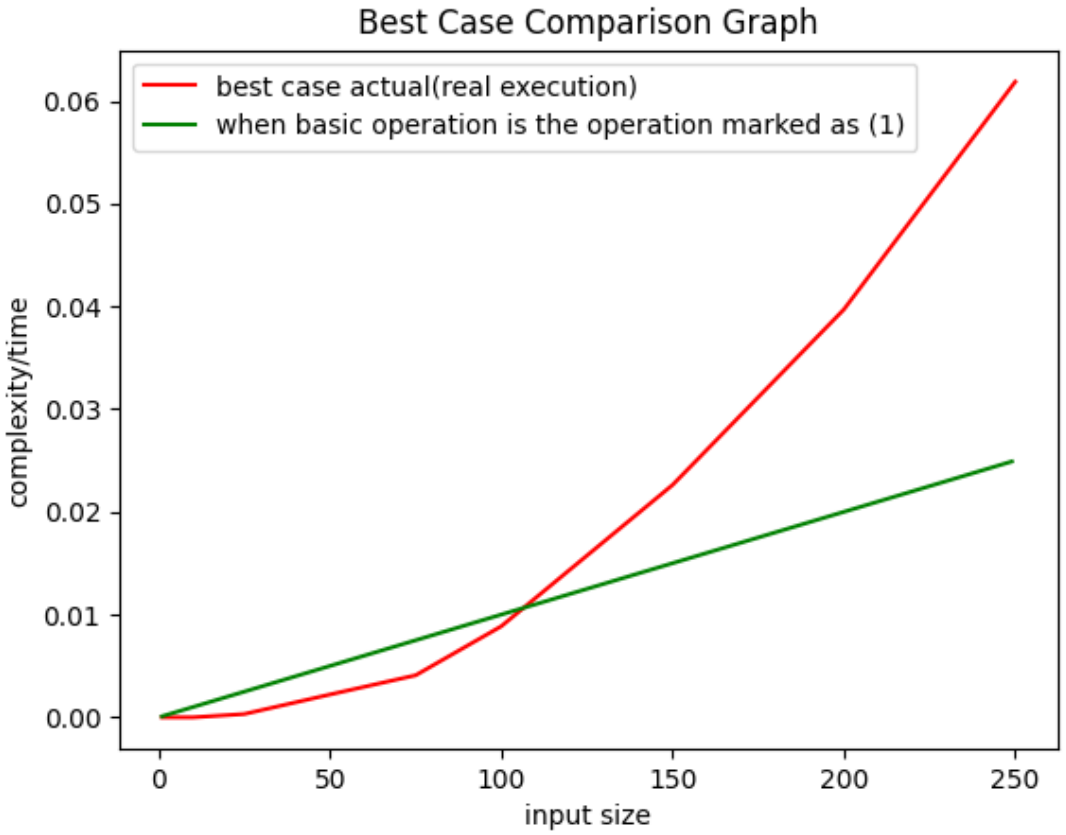
Best Case

Graph of the real execution time of the algorithm



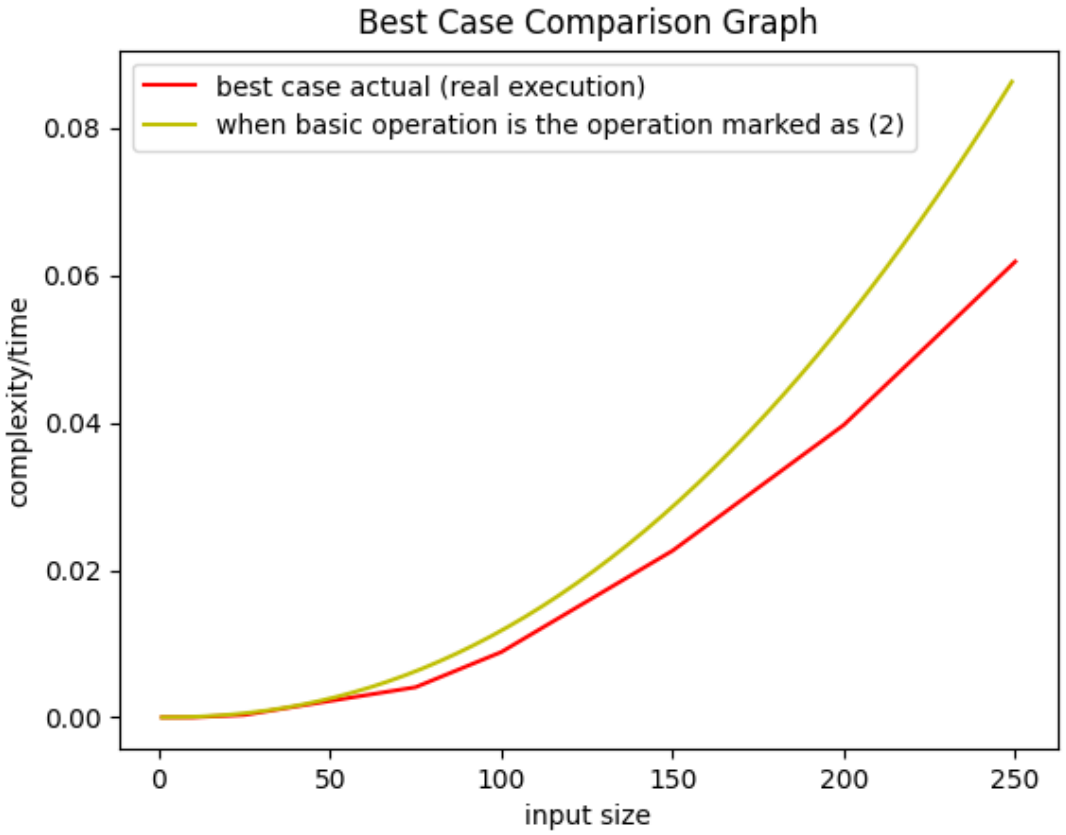
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (1)



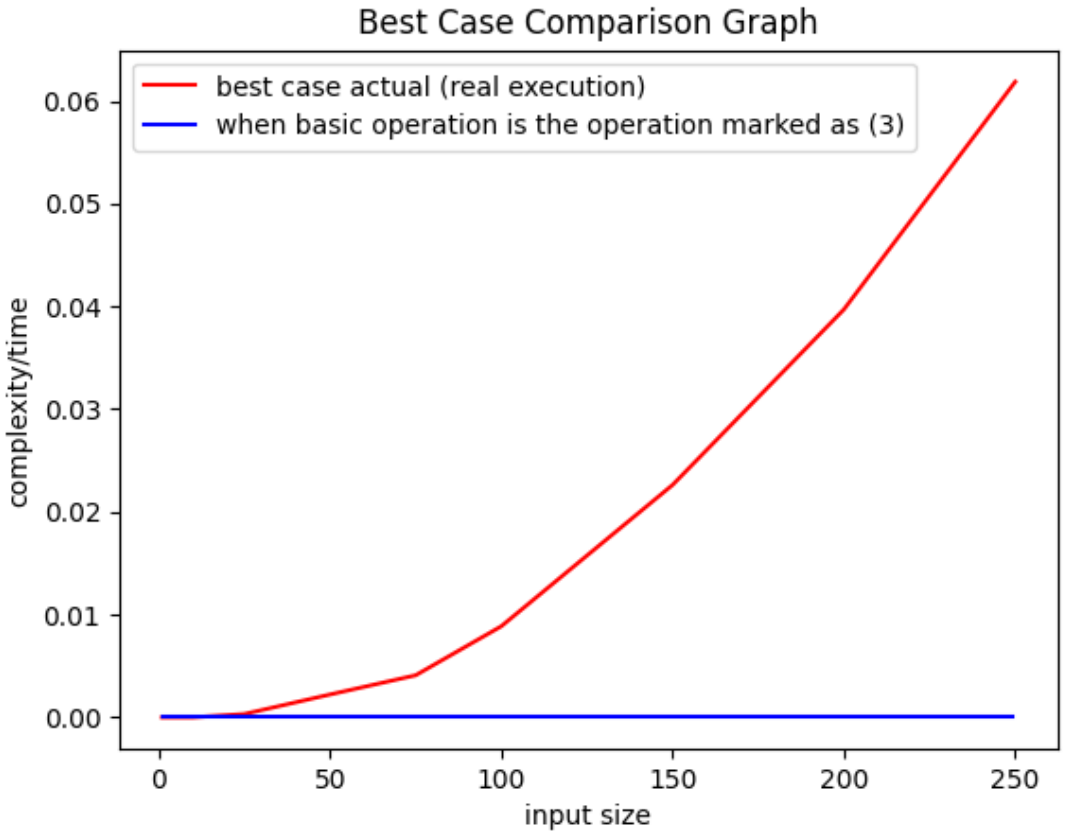
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (2)



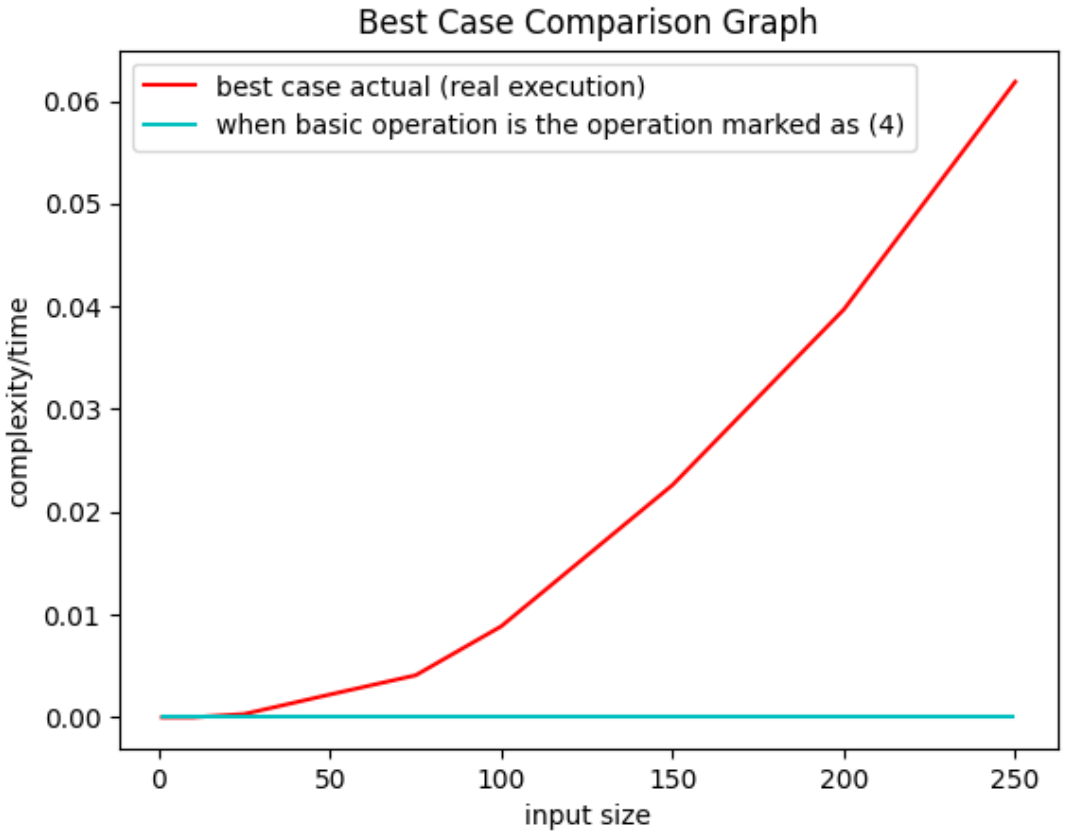
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (3)



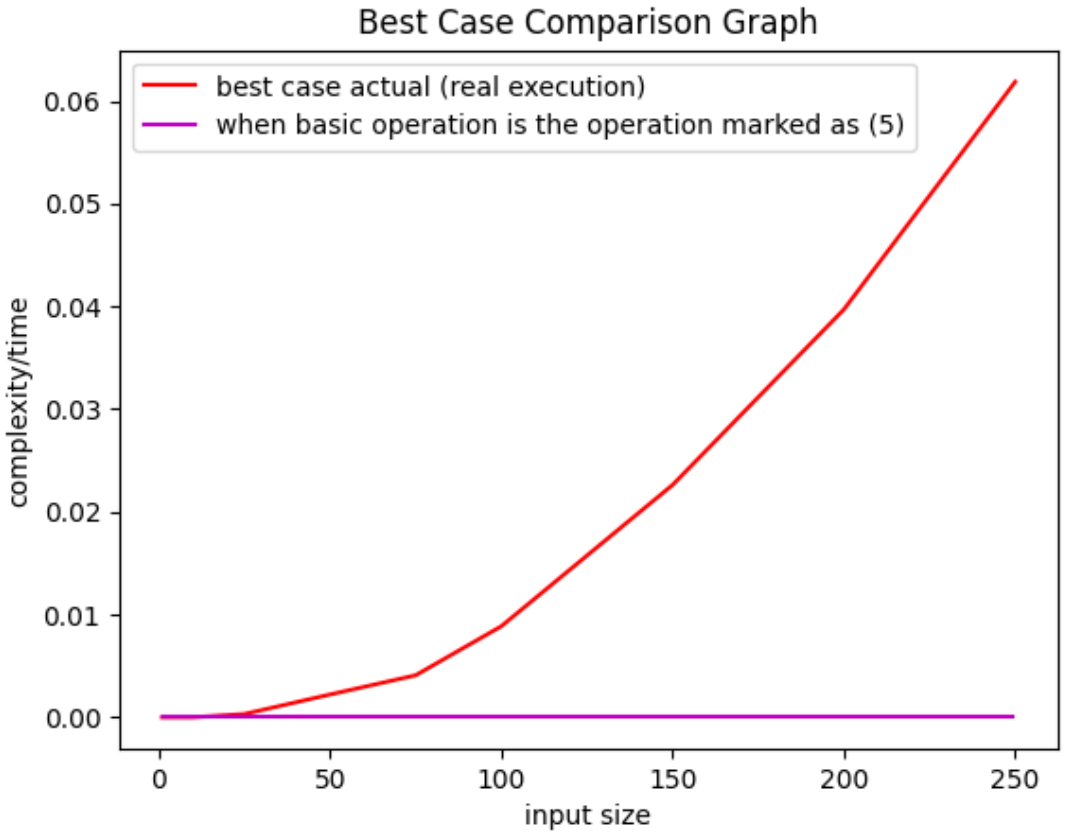
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (4)



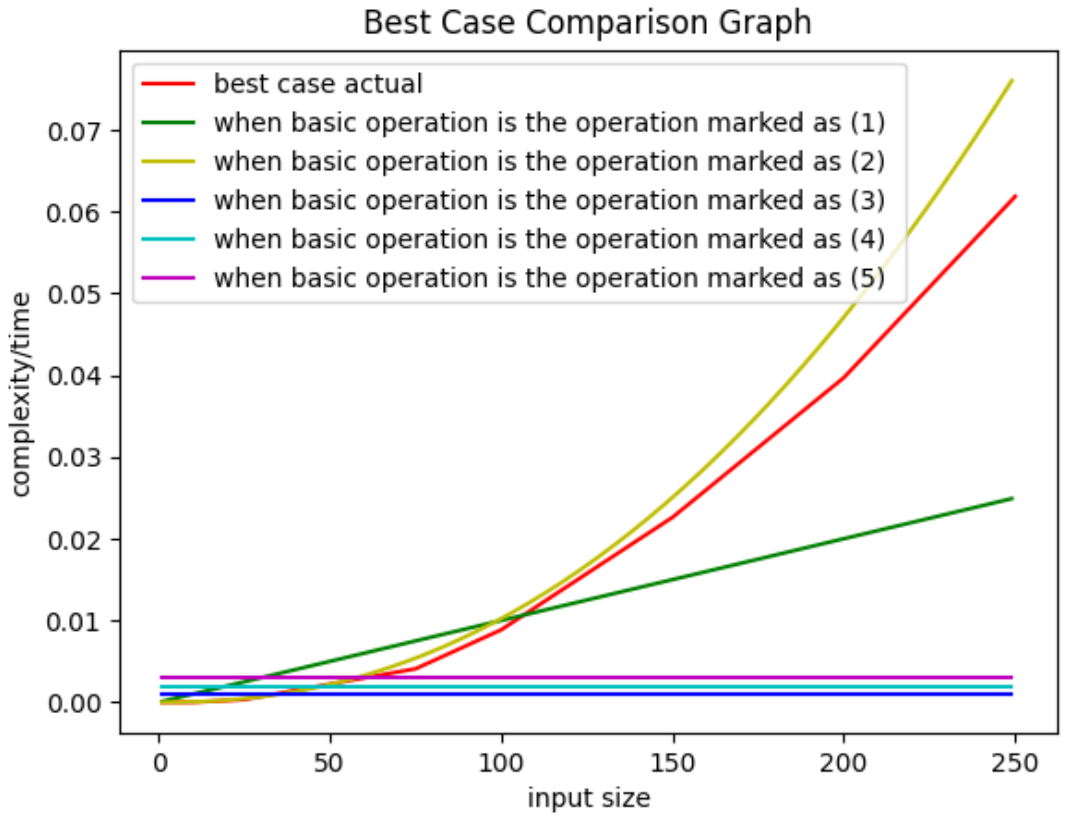
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (5)



Comments

Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354



Complexity when basic operation is the operation marked as (1) : $\theta(f(n) \in \theta(n)$
Complexity when basic operation is the operation marked as (2) : $\theta(f(n) \in \theta(n^2 \log n)$
Complexity when basic operation is the operation marked as (3) : $B(n) = 0$, $\theta(f(n) \in \theta(1)$
Complexity when basic operation is the operation marked as (4) : $B(n) = 0$, $\theta(f(n) \in \theta(1)$
Complexity when basic operation is the operation marked as (5) : $B(n) = 0$, $\theta(f(n) \in \theta(1)$

When I compare theoretical graphs and the one with real execution, it mostly likely the second one (marked as 2). So algorithm performs $n^2 \log n$, when the best case scenario happen.

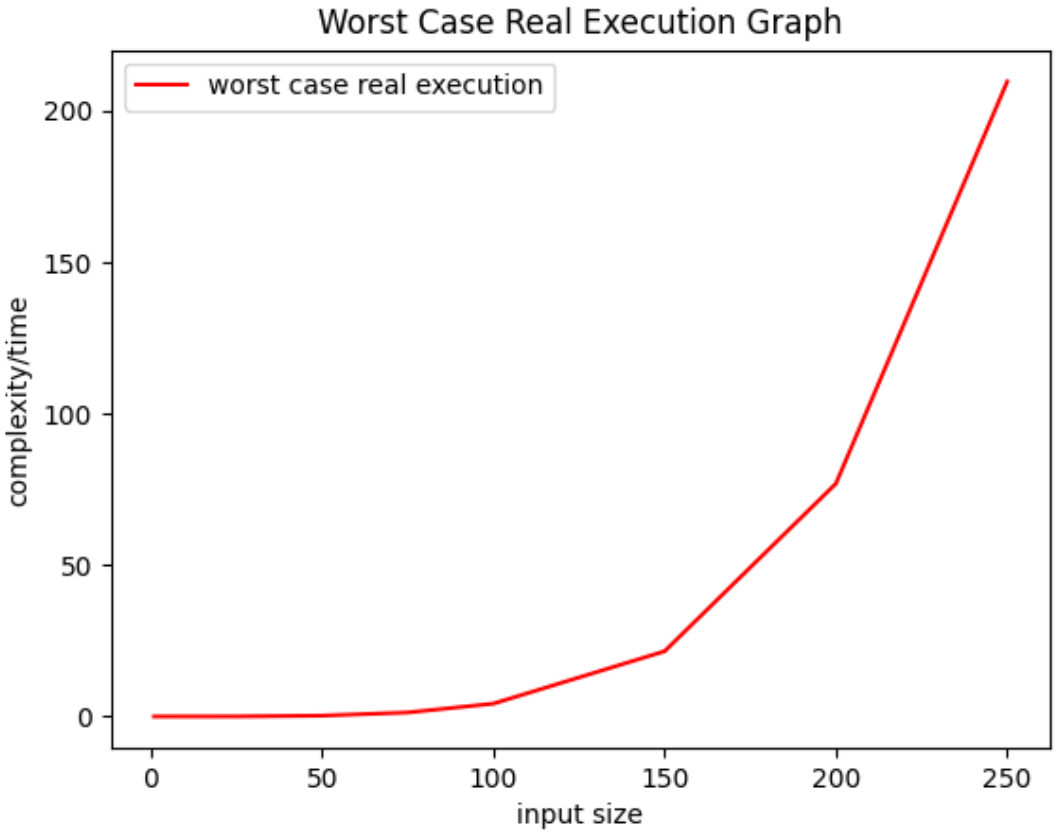
By defining the c1 and c2 constant terms, we could also show that the graph will cover the region between these coefficients.

Important Note : Graphs were drawn using the exact solution . You can see this by examining the written python code (used matplotlib library). For example, $(n^2 + 2n + 1)/2 * \lceil \log_2(n+1) \rceil$ used while drawing the second graph.

Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

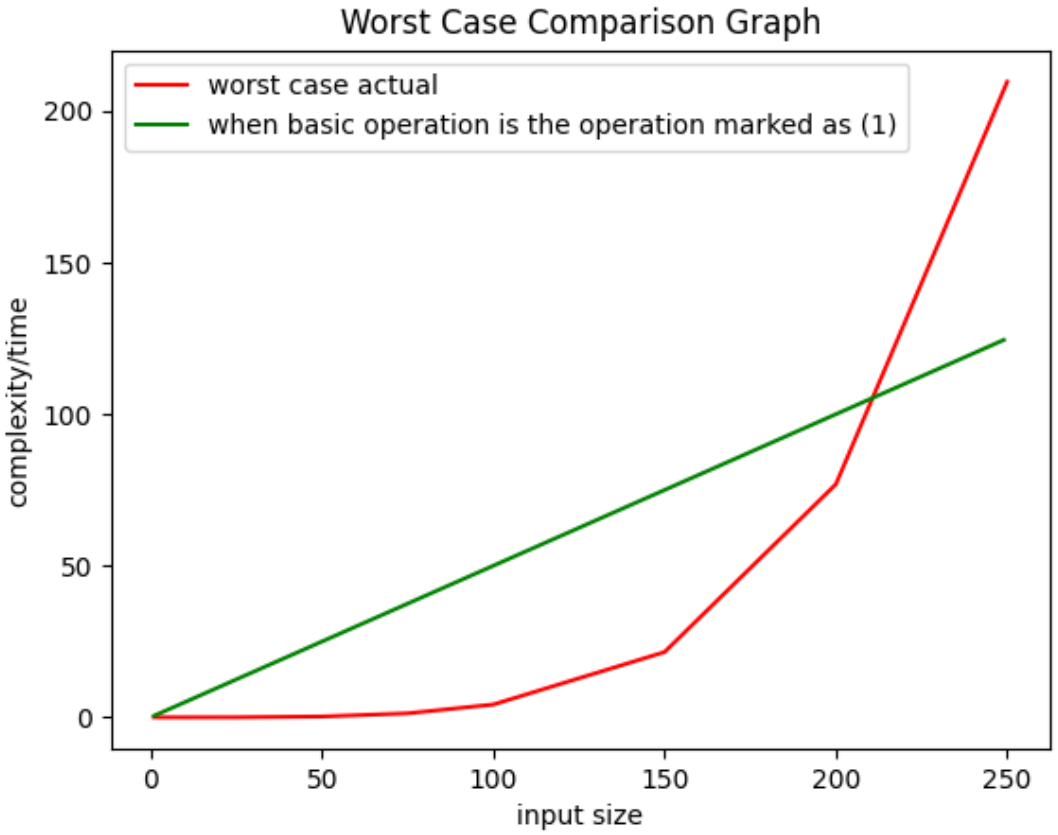
Worst Case

Graph of the real execution time of the algorithm



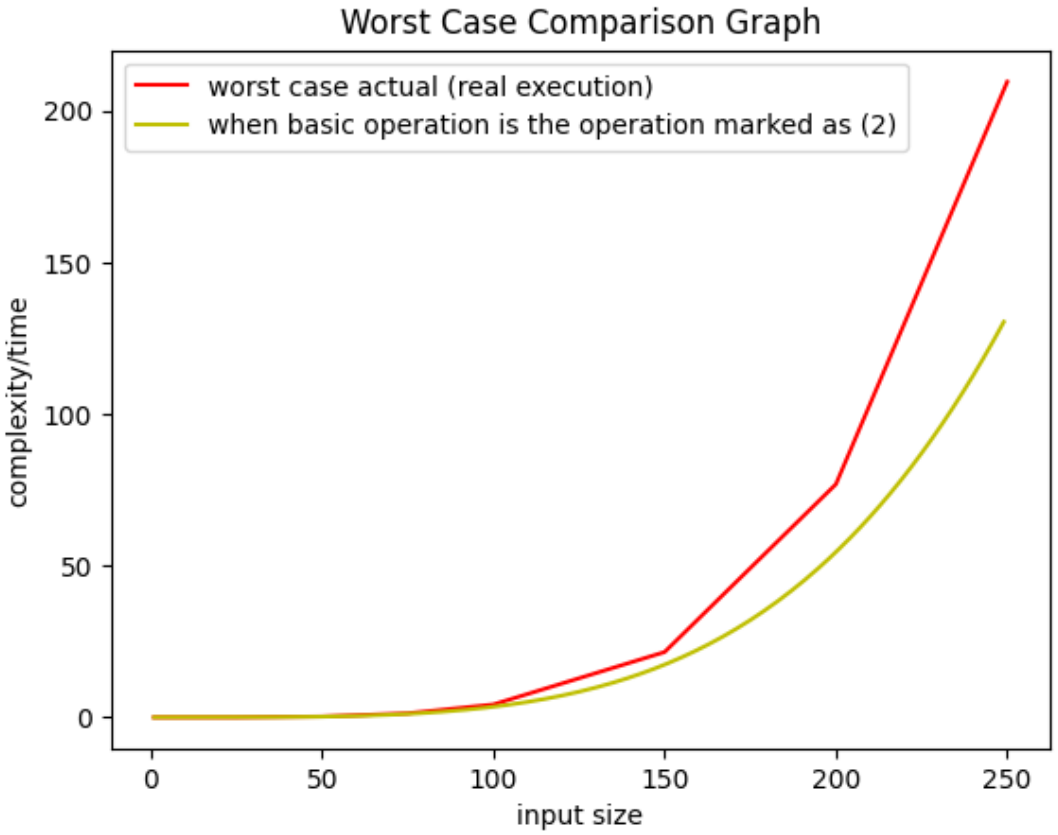
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (1)



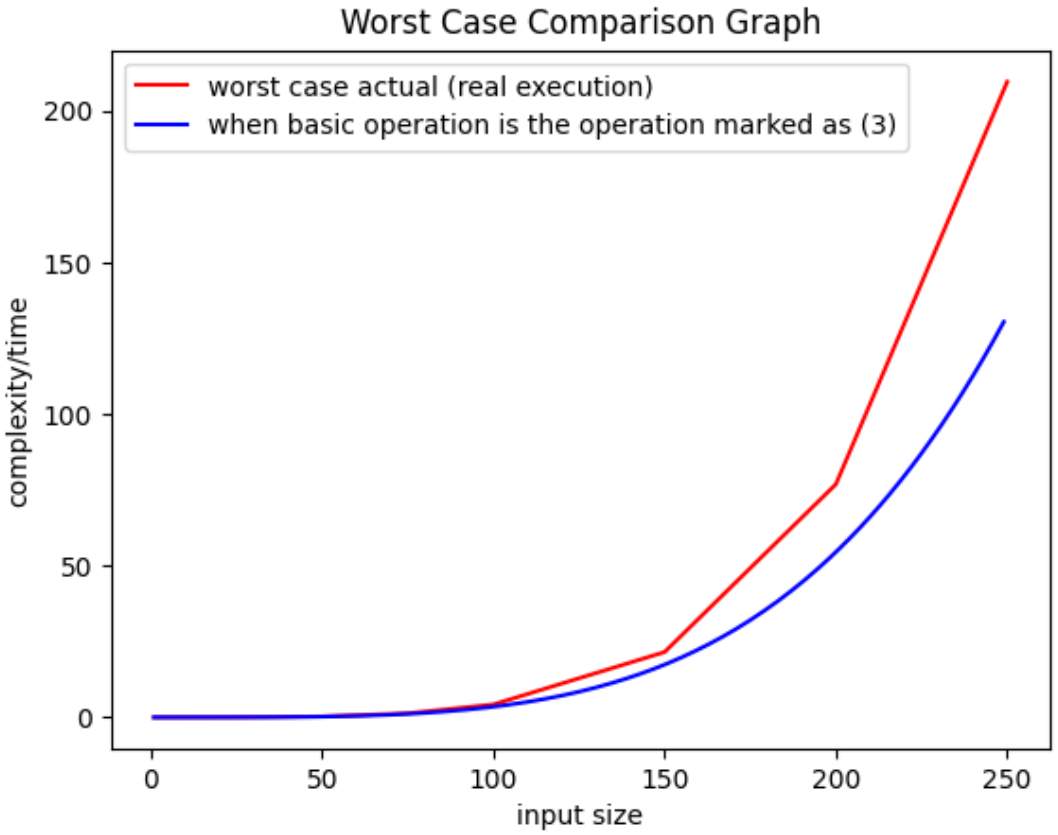
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (2)



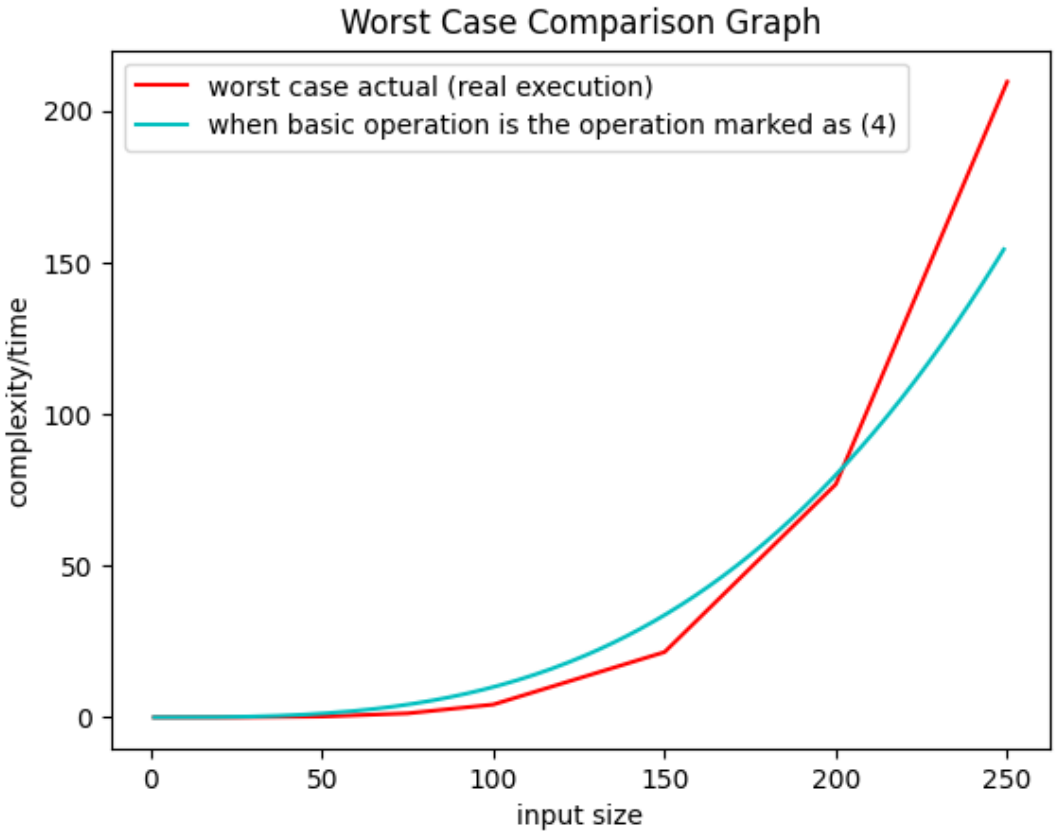
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (3)



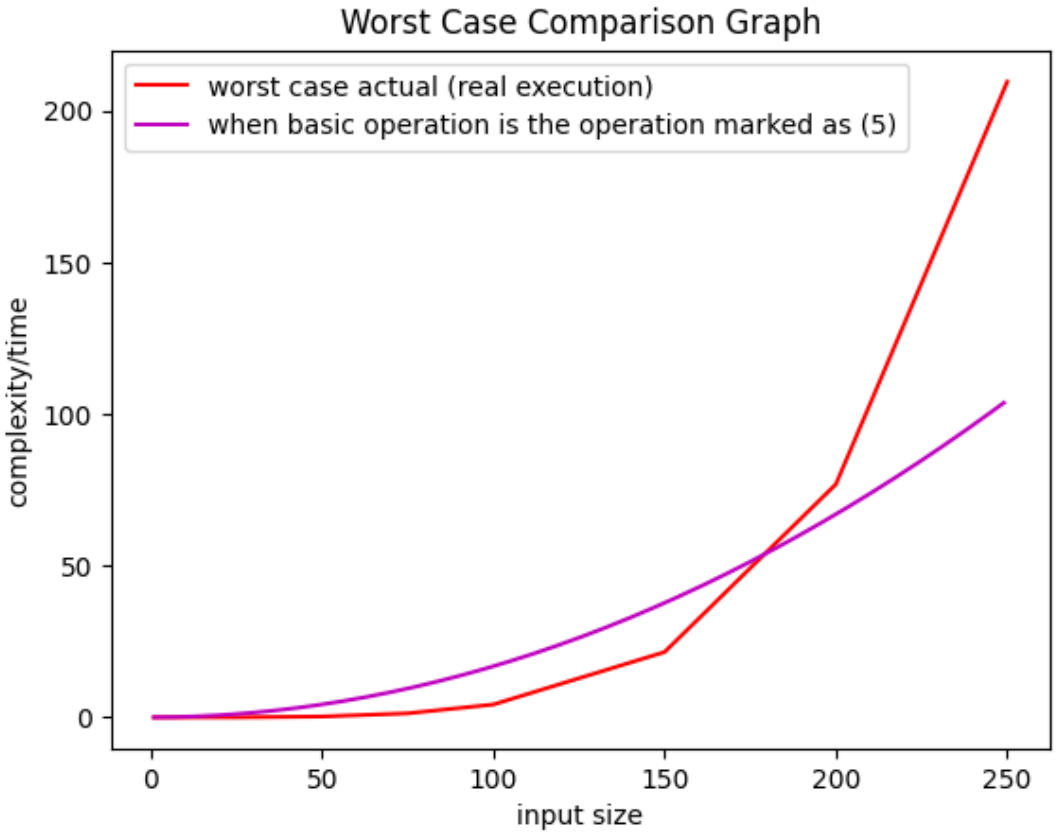
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (4)

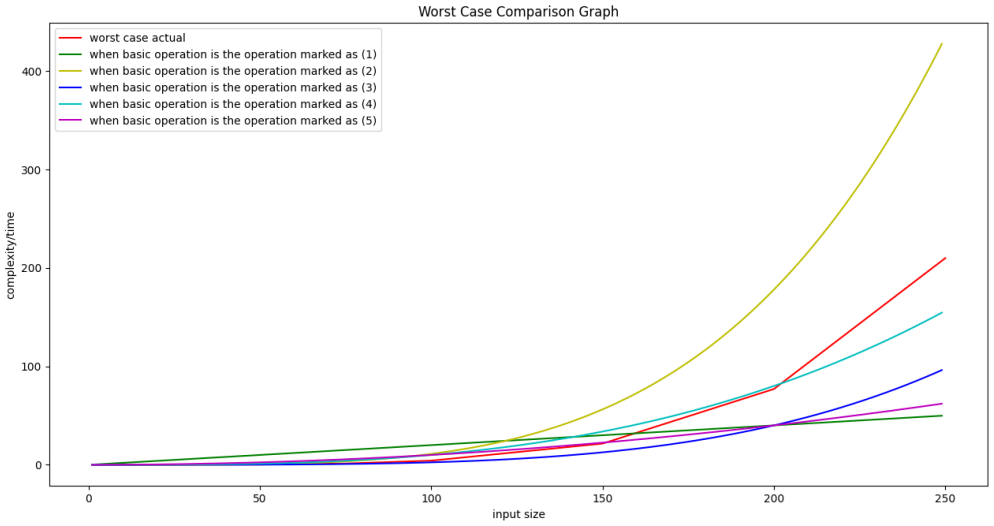


Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (5)



Comments



Worst case complexity when basic operation is the operation marked as (1) : $\theta(f(n) \in \theta(n)$
Worst case complexity when basic operation is the operation marked as (2) : $\theta(f(n) \in \theta(n^4)$
Worst case complexity when basic operation is the operation marked as (3) : $\theta(f(n) \in \theta(n^4)$
Worst case complexity when basic operation is the operation marked as (4) : $\theta(f(n) \in \theta(n^3)$
Worst case complexity when basic operation is the operation marked as (5) : $\theta(f(n) \in \theta(n^2)$

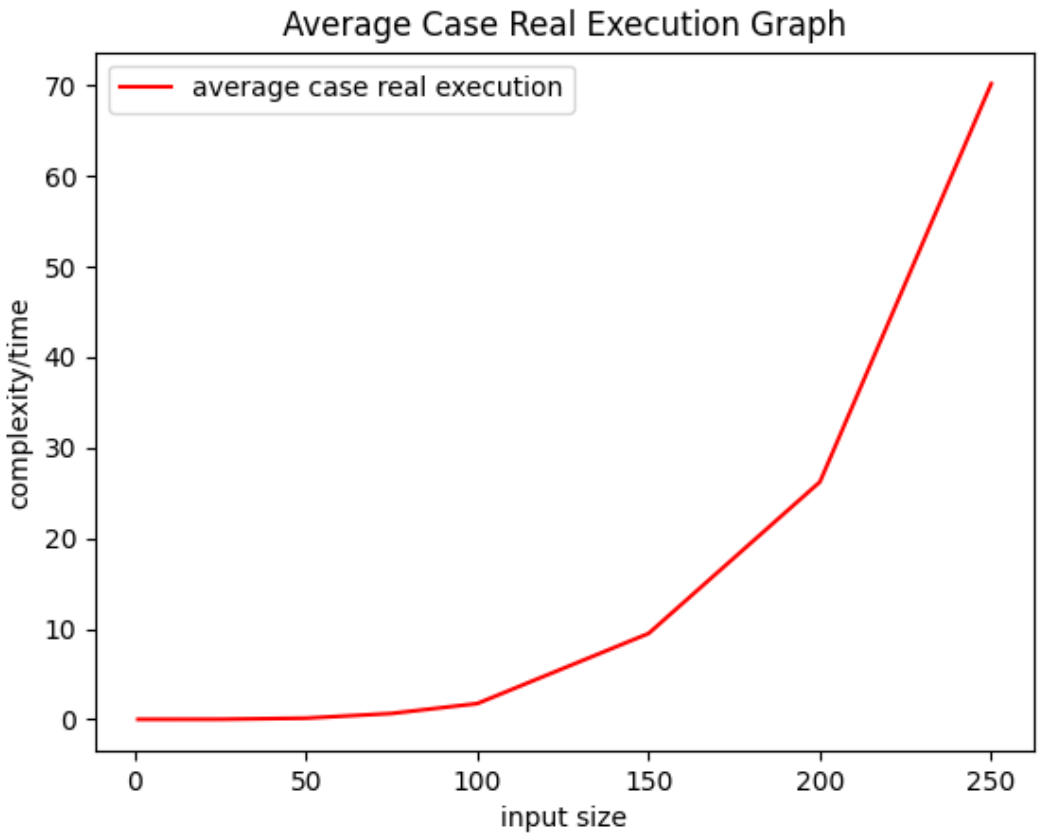
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

When I compare theoretical graphs and the one with real execution, it is most likely the second and third one (marked as (2) and (3)). So the algorithm performs n^4 operations , when the worst case scenario happens. Analyzed complexity(theoretical graphs) is bounded by constant multiples of the actual execution time.

As can be seen in other graphics((1), (4) , (5)), as the input size increases, the graphics overlap and their directions change. Analyzed complexity is not bounded by constant multiples of the actual execution time.

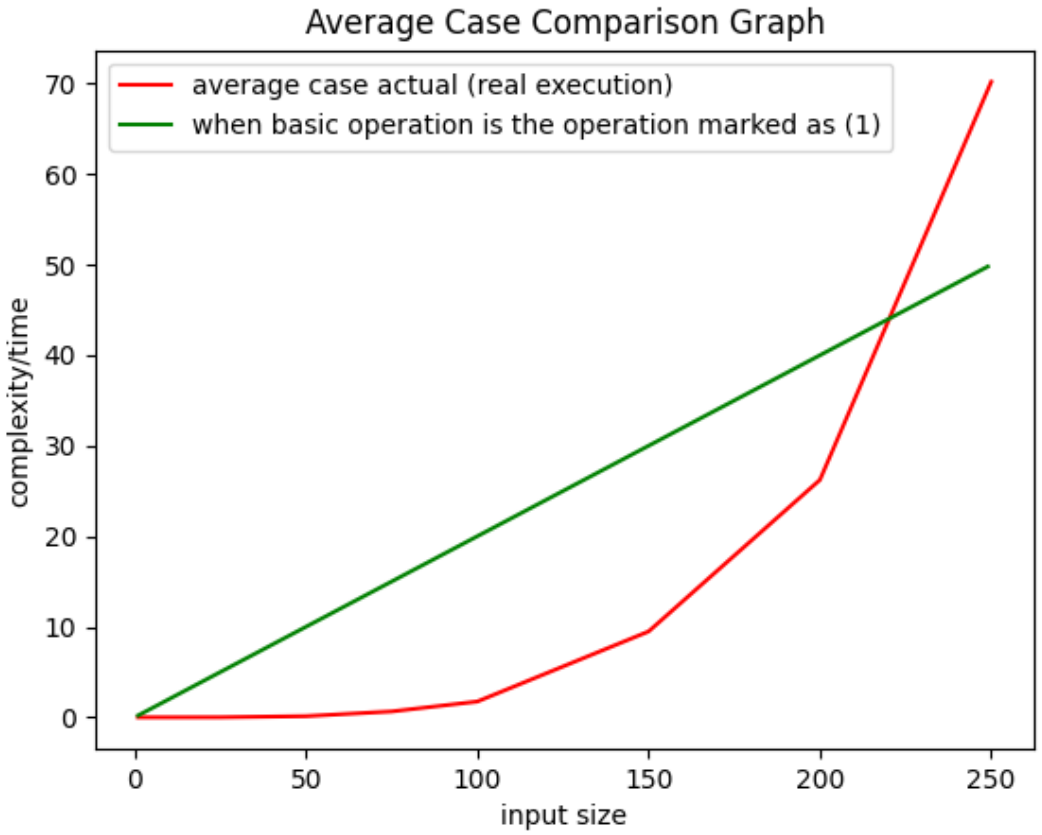
Average Case

Graph of the real execution time of the algorithm



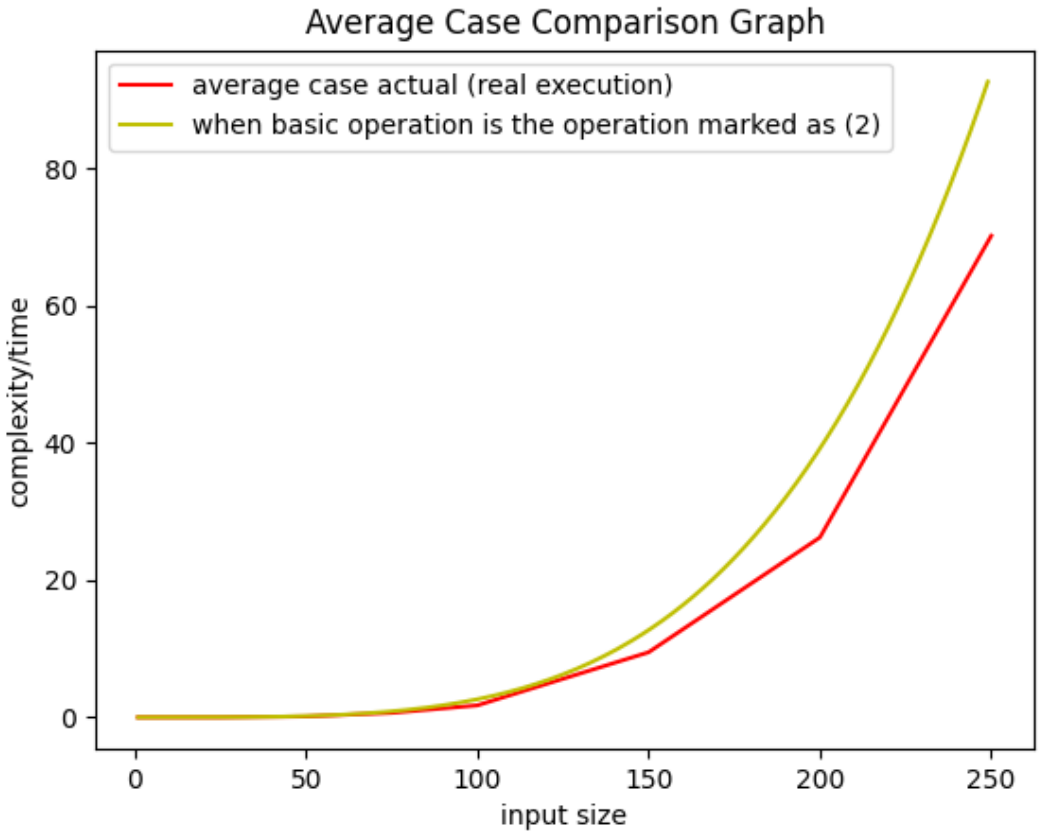
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (1)



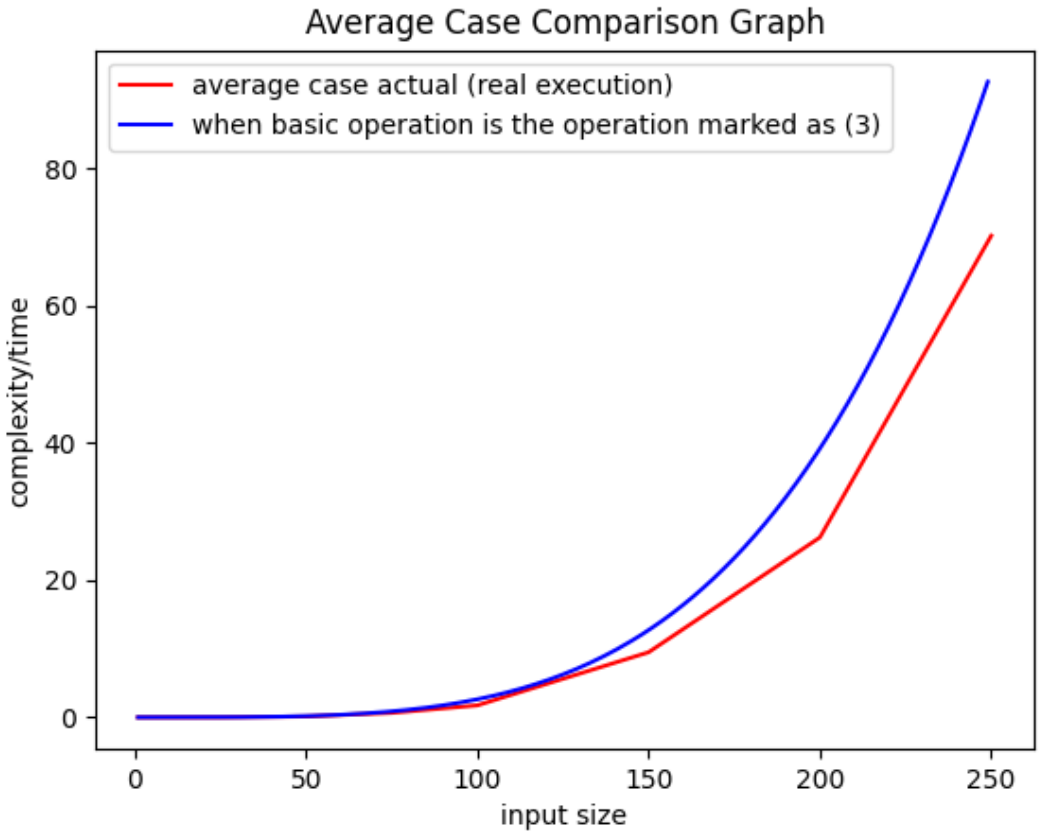
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (2)



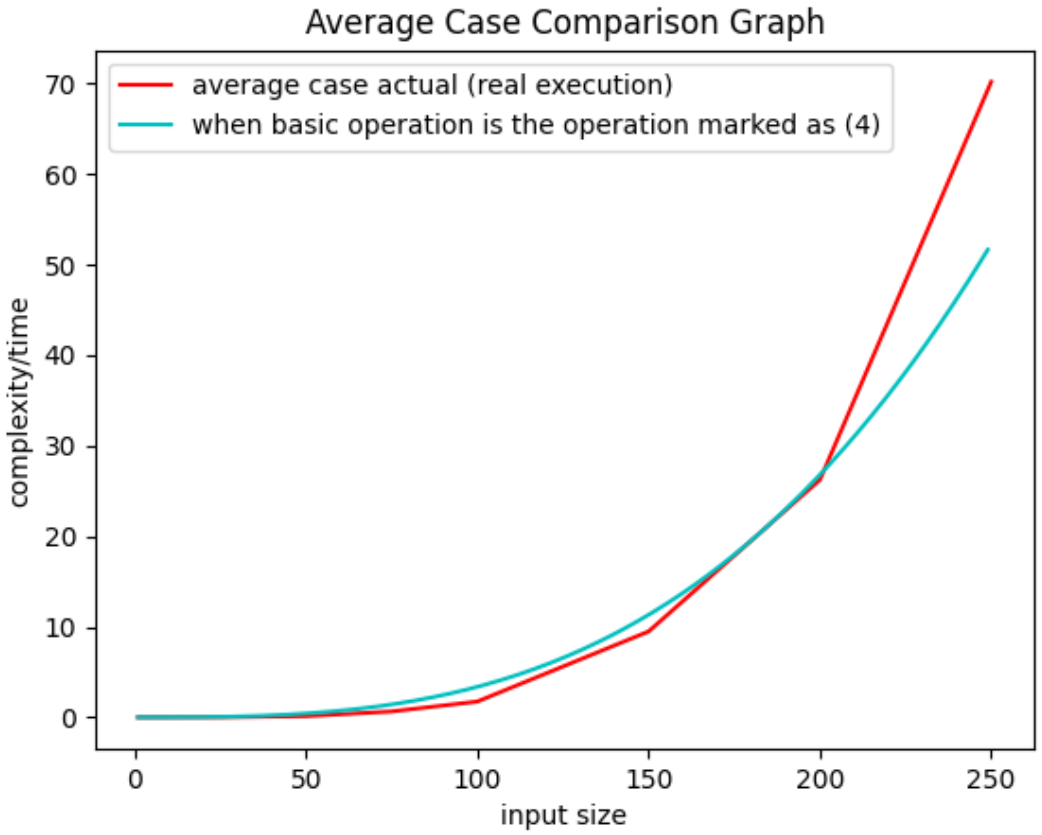
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (3)



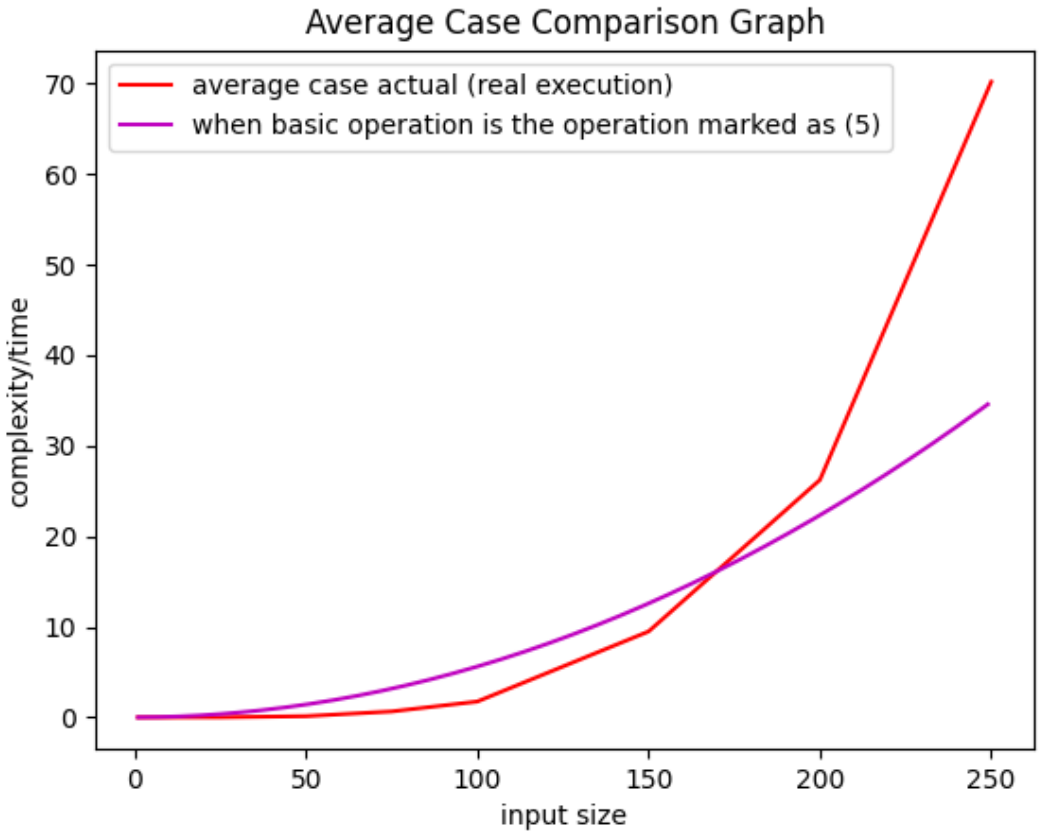
Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (4)

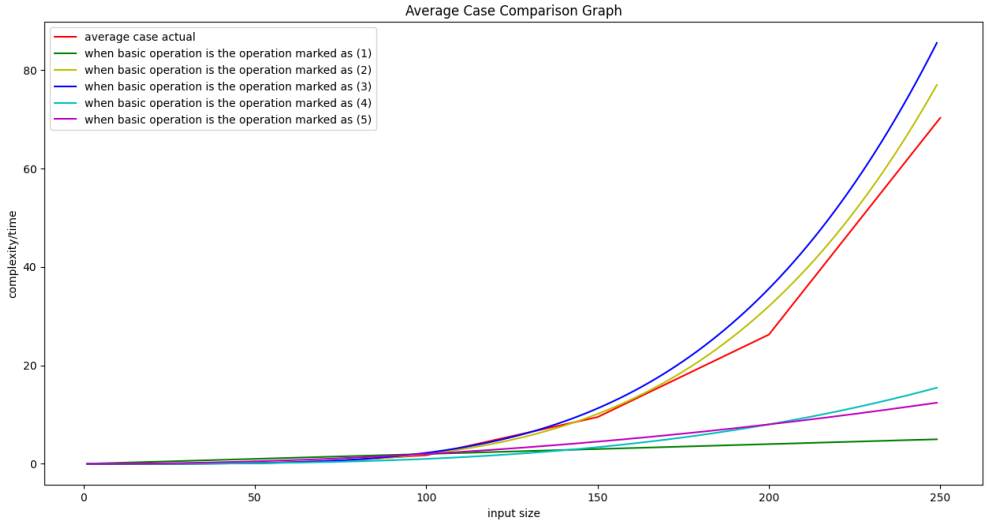


Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

Graph of the theoretical analysis when basic operation is the operation marked as (5)



Comments



Average case complexity when basic operation is the operation marked as (1) : $\theta(f(n) \in \theta(n)$
Average case complexity when basic operation is the operation marked as (2) : $\theta(f(n) \in \theta(n^4)$
Average case complexity when basic operation is the operation marked as (3) : $\theta(f(n) \in \theta(n^4)$
Average case complexity when basic operation is the operation marked as (4) : $\theta(f(n) \in \theta(n^3)$
Average case complexity when basic operation is the operation marked as (5) : $\theta(f(n) \in \theta(n^2)$

Güney Yüksel – 2018400102
Yiğit Sarıoğlu – 2022400354

When I compare theoretical graphs and the one with real execution, it is most likely the second and third one (marked as (2) and (3)). So the algorithm performs n^4 operations , when the average case scenario happens. As can be seen in the graph (2) and (3) , analyzed complexity(theoretical graphs) is bounded by constant multiples of the actual execution time.

As can be seen in other graphics((1), (4) , (5)), as the input size increases, the graphics overlap and their directions change. Analyzed complexity is not bounded by constant multiples of the actual execution time