ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

Table of Contents

ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

THEORETICAL ANALYSIS

Basic operation is the comparison marked as (1)

There are n elements in the input list

Analyze B(n) -> In all cases the if statement is executed, so there are n comparisons. n is a member of $\Theta(n)$.
Analyze W(n) -> In all cases the if statement is executed, so there are n comparisons. n is a member of $\Theta(n)$
Analyze A(n) -> There is 1/3 probability that the element is 0, 1 or 2. In all cases there are n operations. There are n*1/3 + n*1/3 + n*1/3 = n. n is a member of $\Theta(n)$

In conclusion the best case, worst case and the average case is the same if the basic operation is marked as 1.

Basic operations are the three assignments marked as (2)

There are n elements in the input list.

There are (log(n+1)+1)*(n+1)/2 basic operations inside the first if statement. Log(n+1)+1 comes from the while statement inside the if statement. (N+1)/2 is the expected number of execution of i to n-1 for loop. There are n-i operations in each for statement. i can have value 0 to n-1, where the possibility of each is 1/n. If we add up all these values, the expected number of executions is n+1/2.

There are n^2 (log(n+1)+1) basic operations inside the second if statement. n^2 comes from n to 1 and 1 to n for loops. Log(n+1)+1 comes from the while loop inside the second if statement.

There are n*(n+1)*(2n+1)/6 basic operations inside the third if statement. For t3 = 1 there are 1 basic operations, for t3 = 2 there are 4 basic operations …. We should sum these values up to t3=n in order to find the number of basic operations inside the third if statement. This sum equals to $(1+4+9+\ldots+n^2)$. This sum has a closed form which is n*(n+1)*(2n+1)/6.

Analyze B(n) -> The least number of operations are inside the first if statement. If all elements in the list is 0, the program executes only the first if statement. The number of elements in the list is n, so there are n*(log(n+1) + 1)*(n+1)/2 basic operations in the best case. n*(log(n+1)+1)*(n+1)/2 is a member of $\Theta(n^2\log(n))$

Analyze W(n) -> The maximum number of operations are inside the third if statement. If all elements in the list is 2, the program executes only the third if statement. The number of elements in the list is n, so there are n*n*(n+1)*(2n+1)/6 basic operations in the worst case. n*n*(n+1)*(2n+1)/6 is a member of $\Theta(n^4)$.

Analyze A(n) -> Each element of the list has 1/3 probability that it is 0, 1 or 2. So the element executes the first if statement with 1/3 probability, the second if statement with 1/3 probability, the third if statement with 1/3 probability. There are n elements in the list so there are $(n/3)*(((\log(n+1)+1)*(n+1)/2) + (n^2(\log(n+1)+1)) + (n*(n+1)*(2n+1)/6))$ basic operations, which is a member of $\Theta(n^4)$.

ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

Basic operation is two assignments marked as (3)

There are n elements in the input list

There are 0 basic operations inside of the first if statement because there are no marked basic operations.

There are $n^2(\log(n+1)+1)$ basic operations inside the second if statement. n^2 comes from n to 1 and 1 to n for loops. Log(n+1)+1 comes from the while loop inside the second if statement.

There are n*(n+1)*(2n+1)/6 basic operations inside the third if statement. For t3 = 1 there are 1 basic operations, for t3 = 2 there are 4 basic operations …. We should sum these values up to t3=n in order to find the number of basic operations inside the third if statement. This sum equals to $(1+4+9+\ldots+n^2)$. This sum has a closed form which is n*(n+1)*(2n+1)/6.

Analyze B(n) -> The least number of operations are inside the first if statement, which is 0. If all elements in the list is 0, the program executes only the first if statement. There are n elements in the list. n*0 = 0 , which is a member of $\Theta(1)$.

Analyze W(n) -> The maximum number of operations are inside the third if statement. If all elements in the list is 2, the program executes only the third if statement. The number of elements in the list is n, so there are n*n*(n+1)*(2n+1)/6 basic operations in the worst case. n*n*(n+1)*(2n+1)/6 is a member of $\Theta(n^4)$.

Analyze A(n) -> Each element of the list has 1/3 probability that it is 0, 1 or 2. So the element executes the first if statement with 1/3 probability, the second if statement with 1/3 probability, the third if statement with 1/3 probability. There are n elements in the list so there are $(n/3)*(0 + (n^2(\log(n+1)+1)) + (n*(n+1)*(2n+1)/6))$ basic operations, which is a member of $\Theta(n^4)$.

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

There are n elements in the input list

There are 0 basic operations inside of the first if statement because there are no marked basic operations.

There are n^2 basic operations inside of the second if statement. p2 increases 1 to n, so there are n basic operations for each t2. There are n t2 values so there are n*n operations inside of the second if statement.

There are n basic operations inside of the third if statement. t3 increases 1 to n, which implies n basic operations.

Analyze B(n) -> The least number of operations are inside the first if statement, which is 0. If all elements in the list is 0, the program executes only the first if statement. There are n elements in the list. n*0 = 0 , which is a member of $\Theta(1)$.

ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

Analyze W(n) -> The maximum number of operations are inside the second if statement. If all elements in the list is 1, the program executes only the second if statement. The number of elements in the list is n, so there are n*n^2 basic operations in the worst case, which is n^3. n^3 is a member of Θ(n^3).

Analyze A(n) -> Each element of the list has 1/3 probability that it is 0, 1 or 2. So the element executes the first if statement with 1/3 probability, the second if statement with 1/3 probability, the third if statement with 1/3 probability. There are n elements in the list so there are (n/3)*(0 + (n^2) + (n))basic operations, which is a member of Θ(n^3).

Basic operation is the assignment marked as (5)

There are n elements in the input list

There are (n+1)/2 basic operations inside of the first if statement. (N+1)/2 is the expected number of executions of i to n-1 for loop. There are n-i operations in each for statement. i can have value 0 to n-1, where the possibility of each is 1/n. If we add up all these values, the expected number of executions is (n+1)/2.

There are 0 basic operations inside of the second if statement because there are no marked basic operations.

There are 0 basic operations inside of the third if statement because there are no marked basic operations.

Analyze B(n) -> The least number of operations are inside the second or third if statement, which is 0. If all elements in the list is 1 or 2, the program executes only the second and third if statements. There are n elements in the list. n*0 = 0 , which is a member of Θ(1).

Analyze W(n) -> The maximum number of operations are inside the first if statement. If all elements in the list is 0, the program executes only the first if statement. The number of elements in the list is n, so there are n*(n+1)/2 basic operations in the worst case. n*(n+1)/2 is a member of Θ(n^2).

Analyze A(n) -> Each element of the list has 1/3 probability that it is 0, 1 or 2. So the element executes the first if statement with 1/3 probability, the second if statement with 1/3 probability, the third if statement with 1/3 probability. There are n elements in the list so there are (n/3)*((n+1)/2 + 0 + 0)basic operations, which is a member of Θ(n^2).

IDENTIFICATION OF BASIC OPERATION(S)

The operations marked as 2 should be the basic operations because the most important operations are the operations marked as 2 and it contributes most to the total execution time. It also performs repeatedly.

ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

REAL EXECUTION

Best Case

N Size	Time Elapsed
1	0ms
5	0ms
10	0ms
25	1.02ms
50	2.03ms
75	3.96ms
100	7.00ms
150	20.01ms
200	38.00ms
250	54.05ms

Worst Case

N Size	Time Elapsed
1	0ms
5	0ms
10	11.12ms
25	13.44ms
50	174.72ms
75	1008.86ms
100	2947.86ms
150	14996.25ms
200	45972.58ms
250	120037.49ms

Average Case

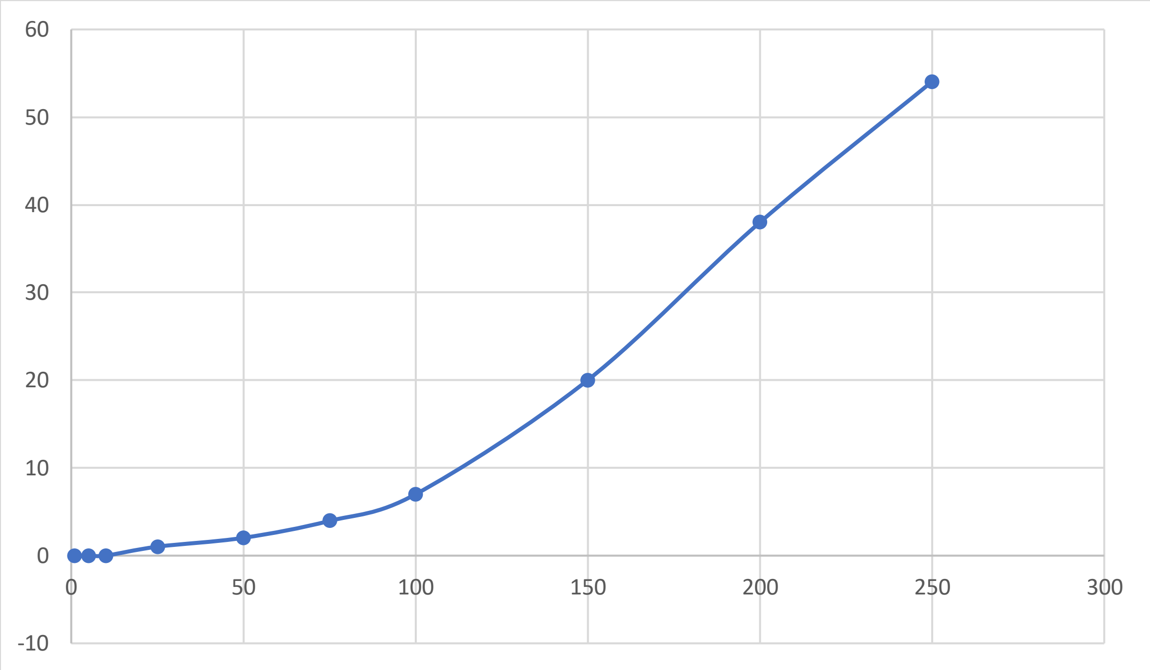
N Size	Time Elapsed
1	0ms
5	0ms
10	2.77ms
25	19.83ms
50	104.47ms
75	479.31ms
100	1476.40ms
150	6517.72ms
200	20252.58ms
250	47412.18ms

ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

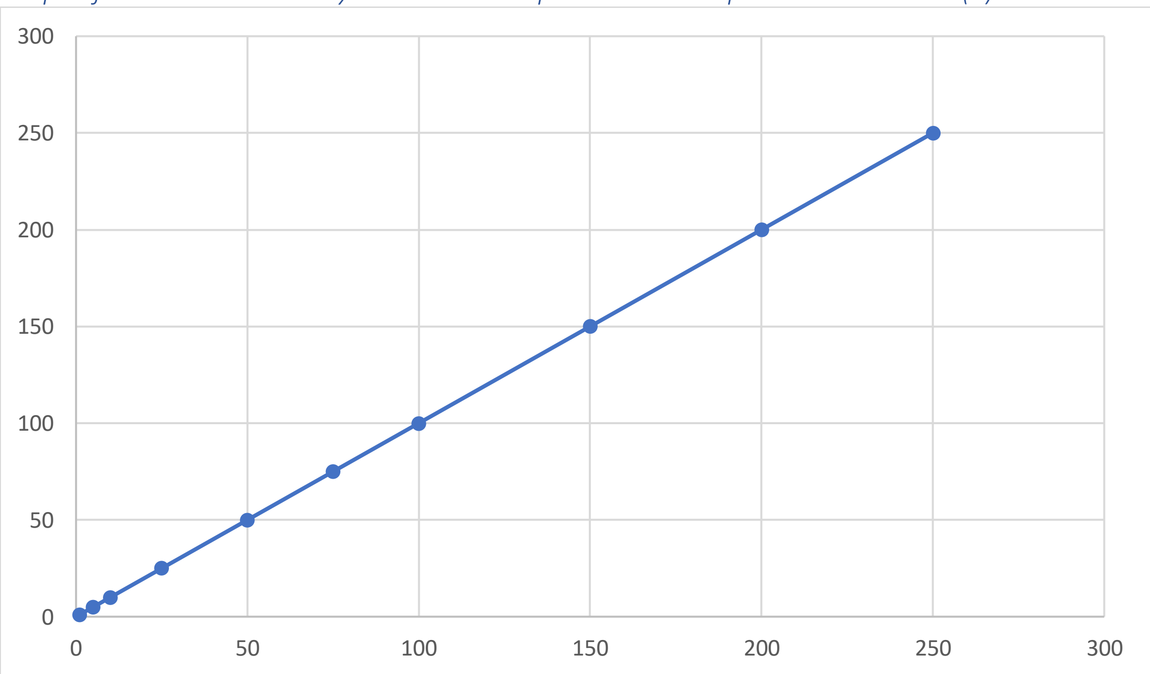
COMPARISON

Best Case

Graph of the real execution time of the algorithm

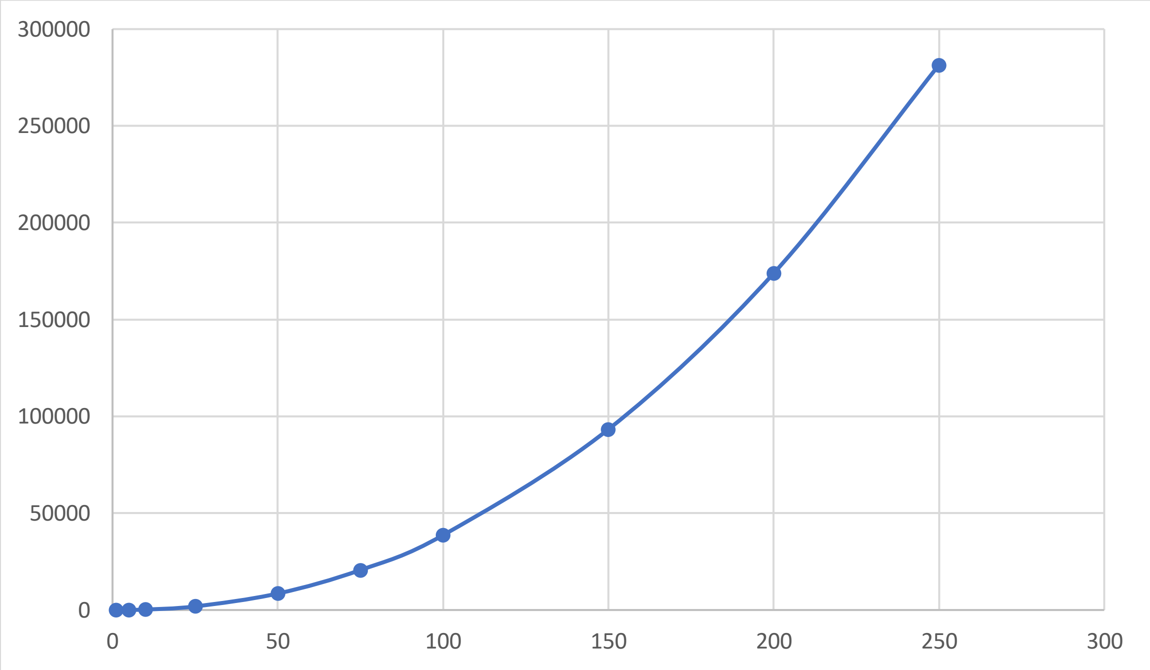


Graph of the theoretical analysis when basic operation is the operation marked as (1)

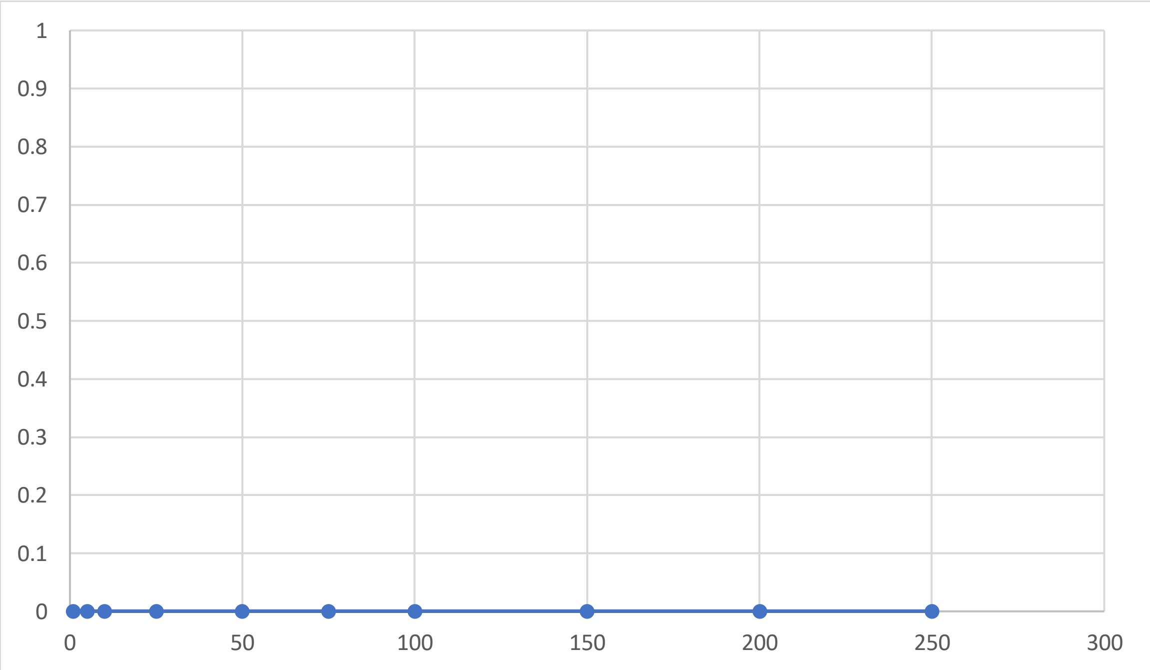


ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

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

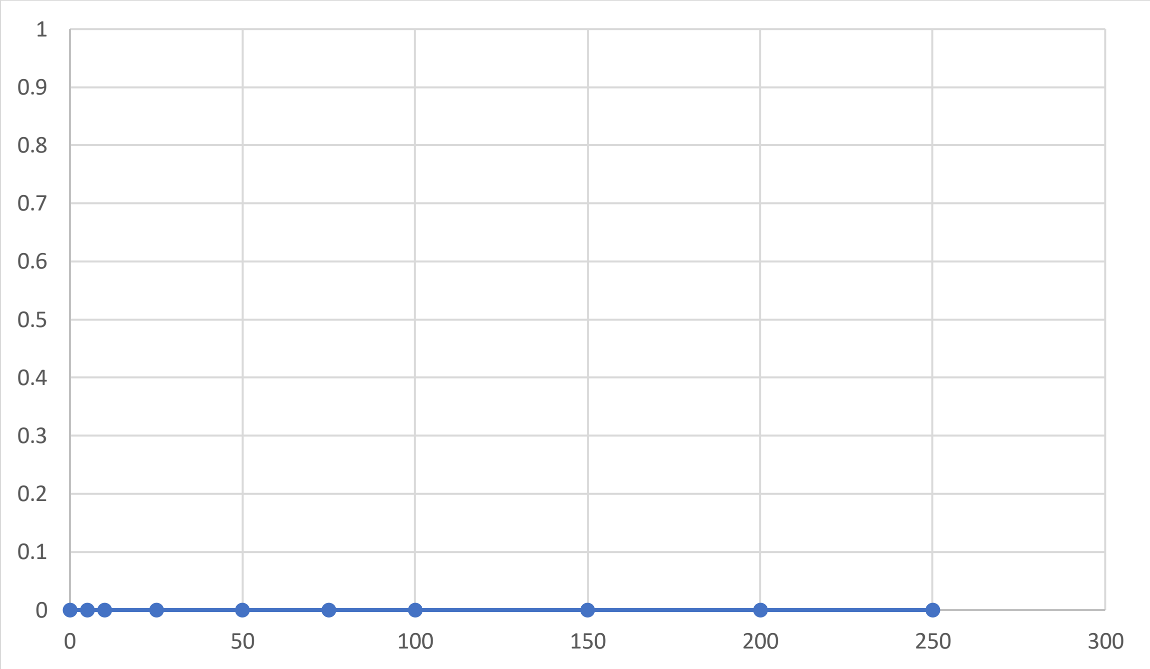


Graph of the theoretical analysis when basic operation is the operation marked as (3)

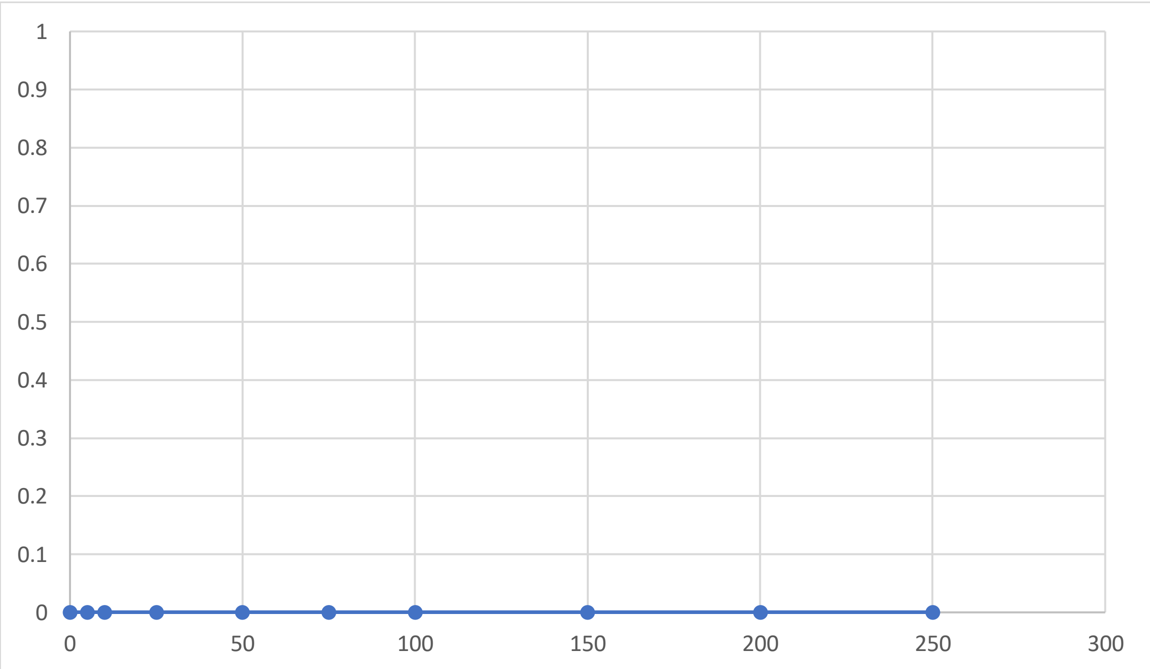


ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

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



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



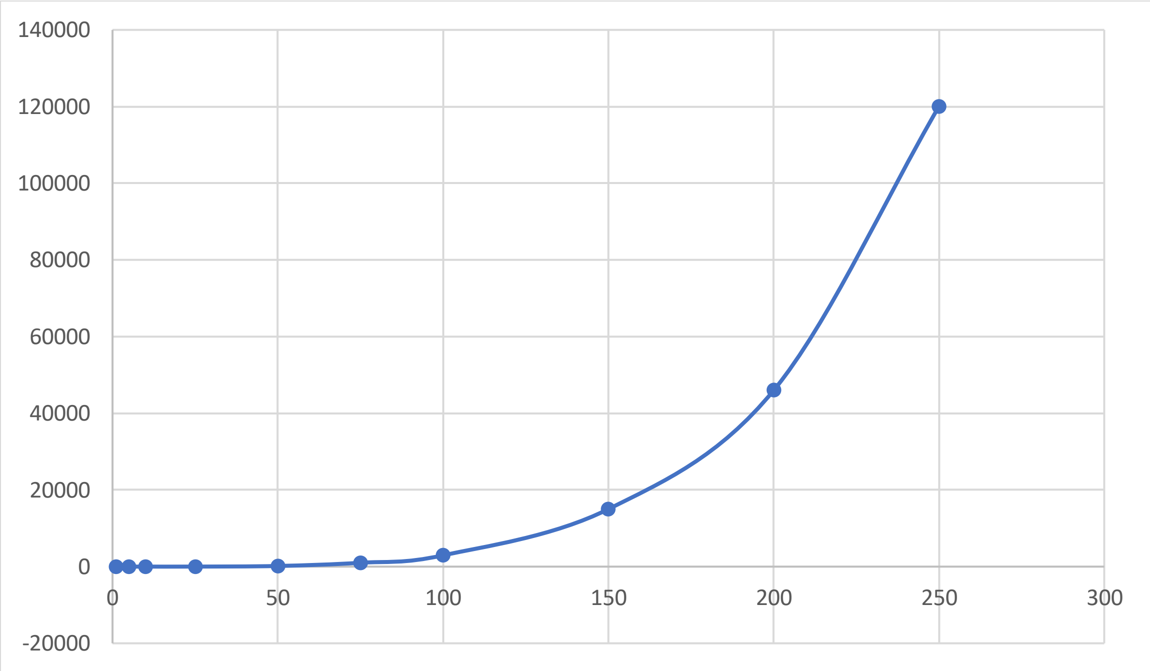
Comments

The graph of real execution times is growing faster than the graph of theoretical analysis when the basic operation is the operation marked as (1) as input size goes to infinity since the real best case complexity of the algorithm is greater than the best case complexity of the algorithm when the basic operation is the operation marked as (1).The 4^{th},5^{th} and 6^{th} graphs have constant y values since the best case complexities for the related theoretical analysis are constant 0. The graph of real execution time and the third graph have similar growth rates.

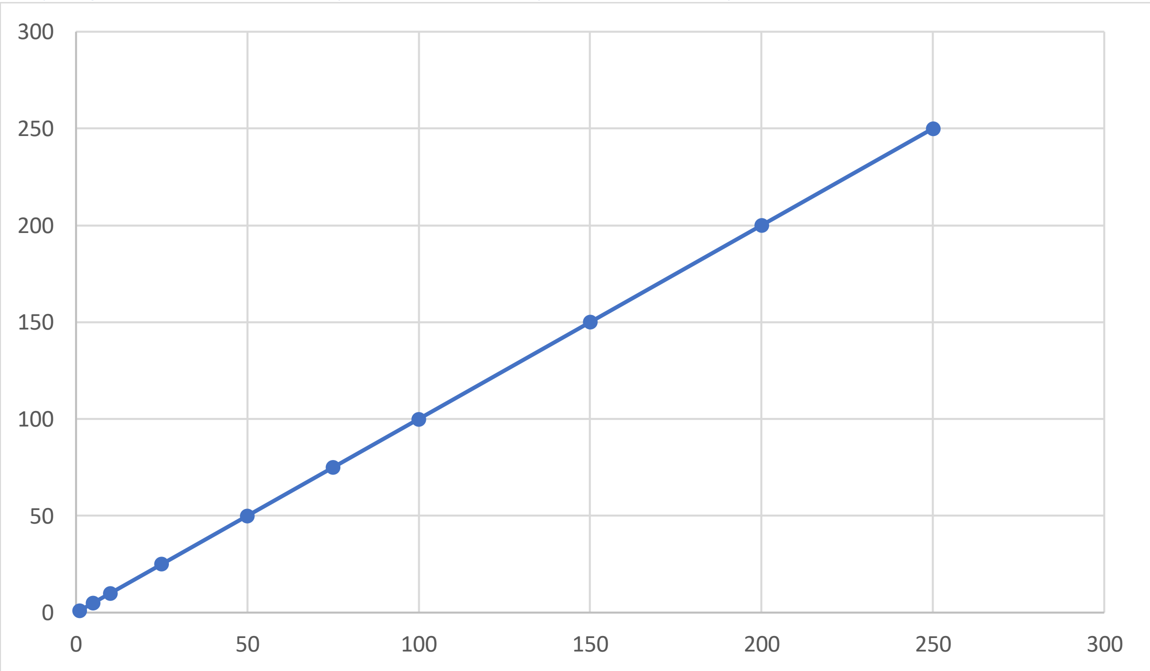
ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

Worst Case

Graph of the real execution time of the algorithm

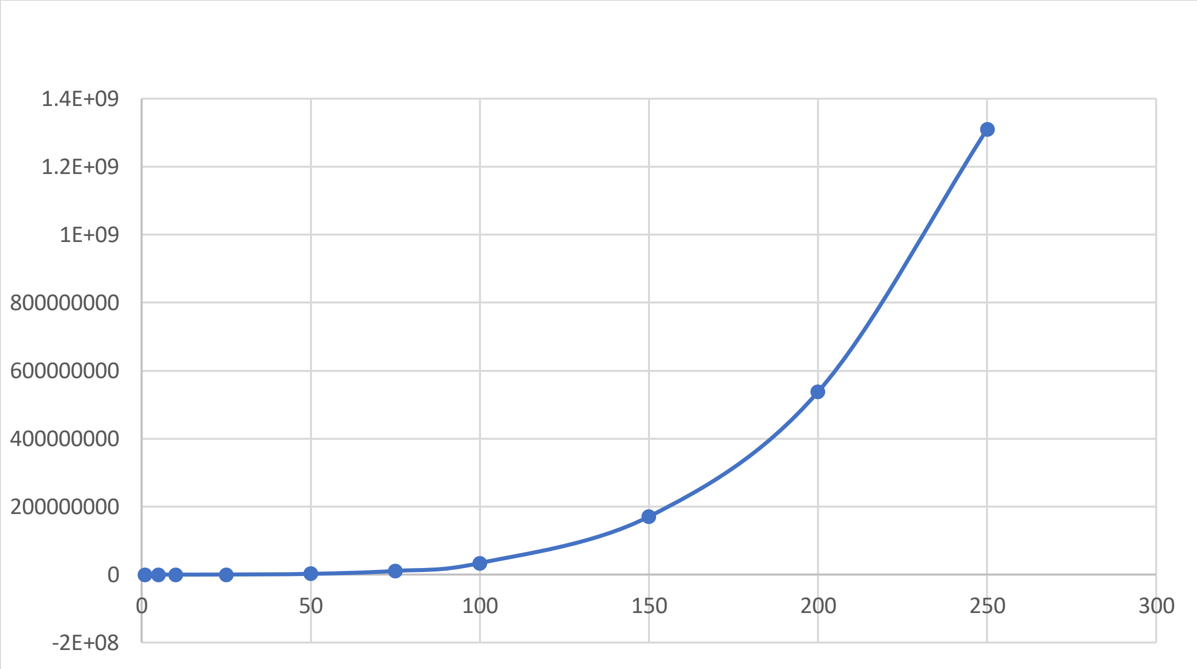


Graph of the theoretical analysis when basic operation is the operation marked as (1)

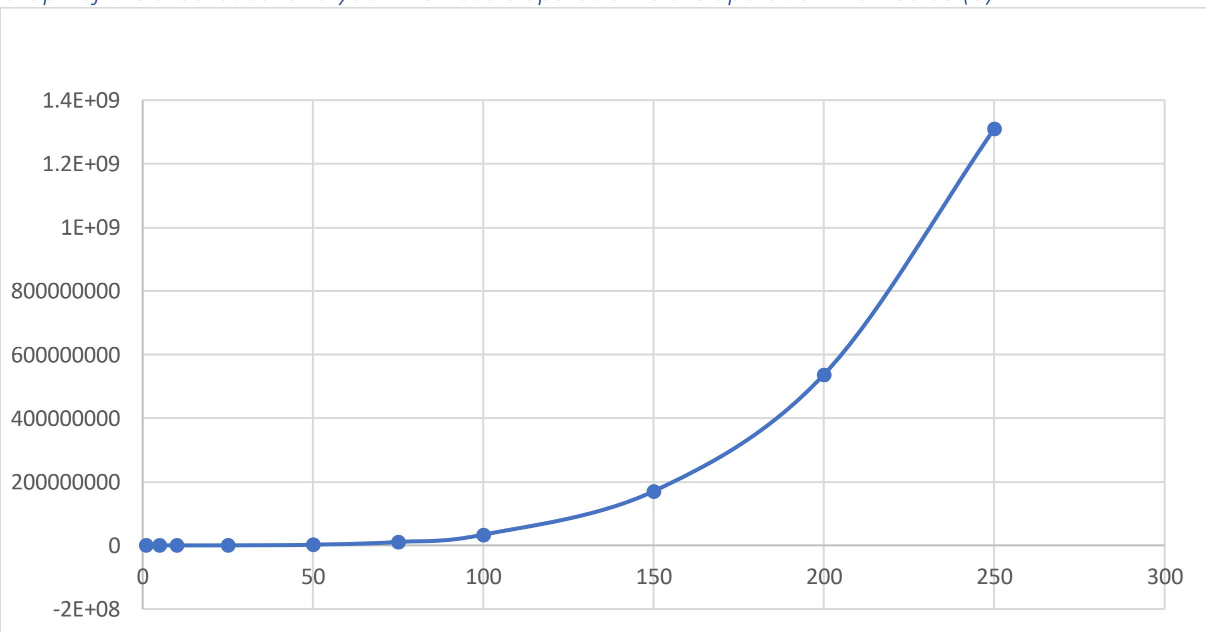


ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

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

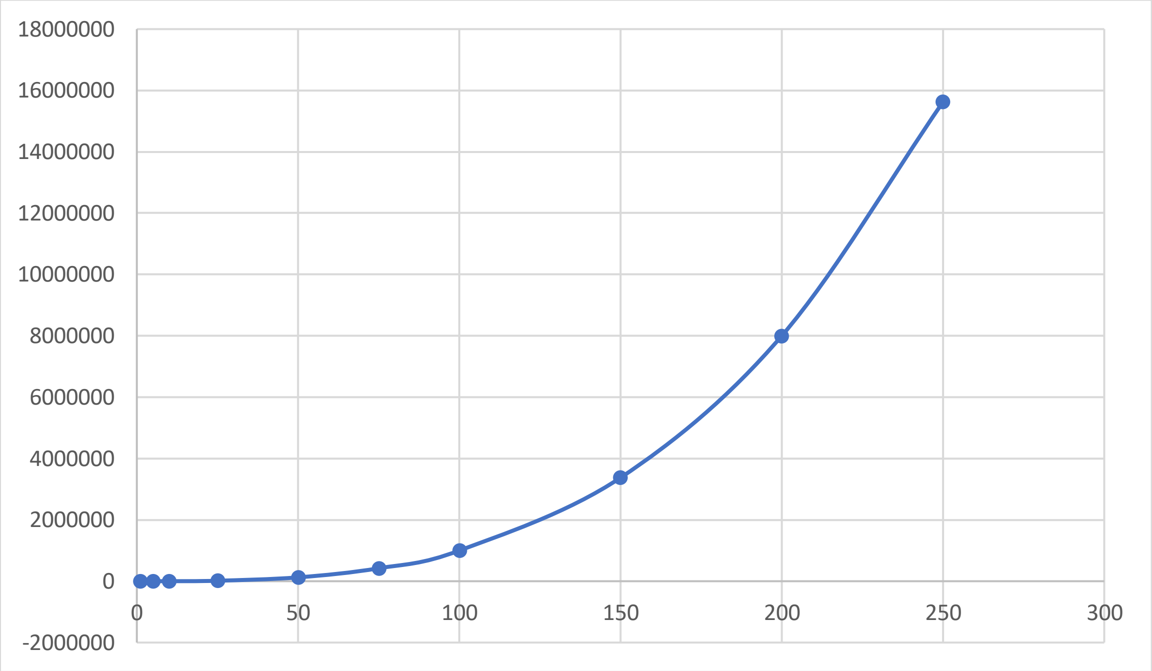


Graph of the theoretical analysis when basic operation is the operation marked as (3)

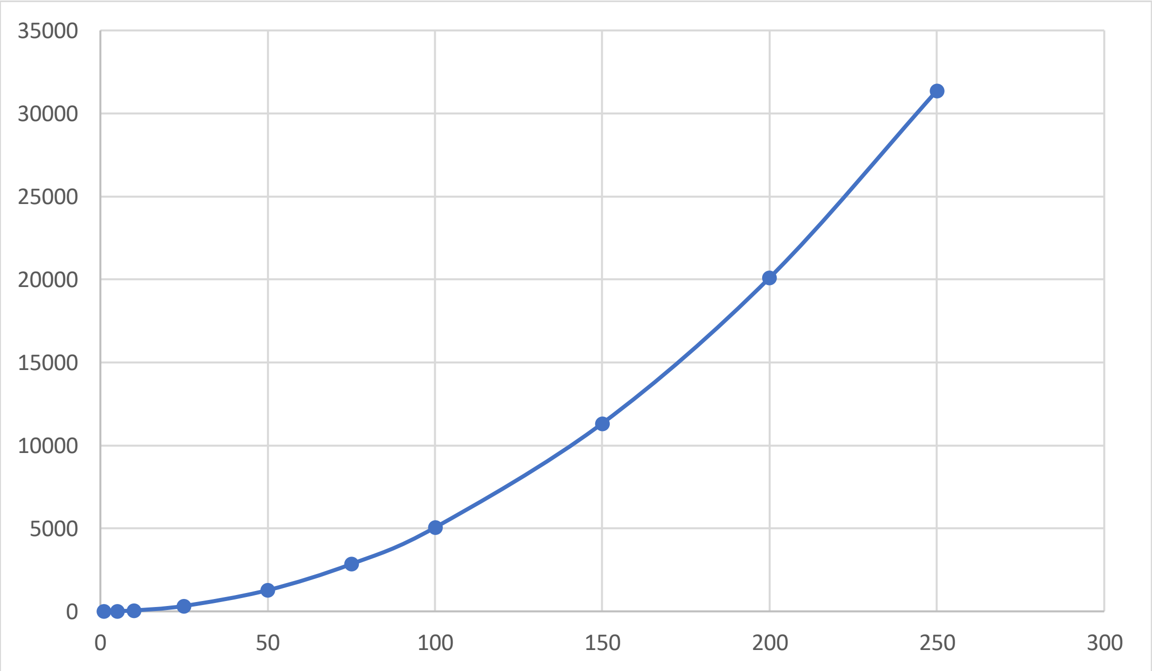


ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

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



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



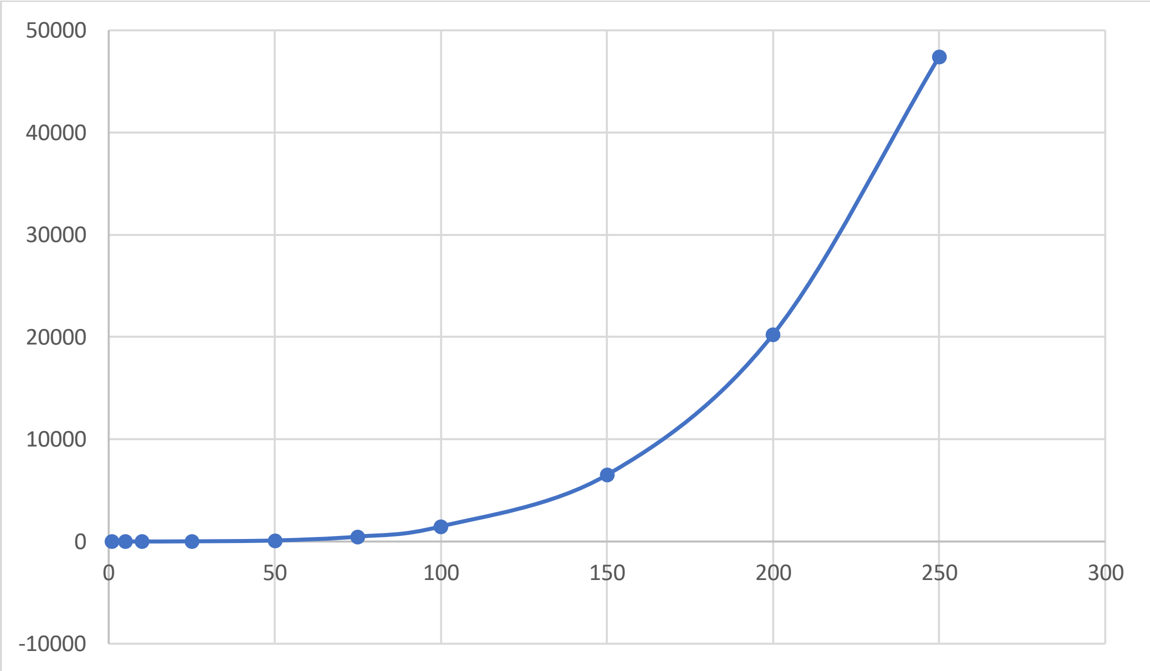
Comments

The graph of real execution times is growing faster than the graph of theoretical analysis when the basic operation is the operation marked as (1) as input size goes to infinity since the real worst case complexity of the algorithm is greater than the worst case complexity of the algorithm when the basic operation is the operation marked as (1).The 3rd and 4th graphs are the same since their related theoretical analysis for the worst case have the same complexity. These two graphs are similar to the real execution time graph. The 5th graph is growing slower than the 3rd graph and the 6th graph is growing slower than the 5th graph as input size goes to infinity. Also, the 5th and 6th graphs have greater growth rates than the second graph as the input size goes to infinity.

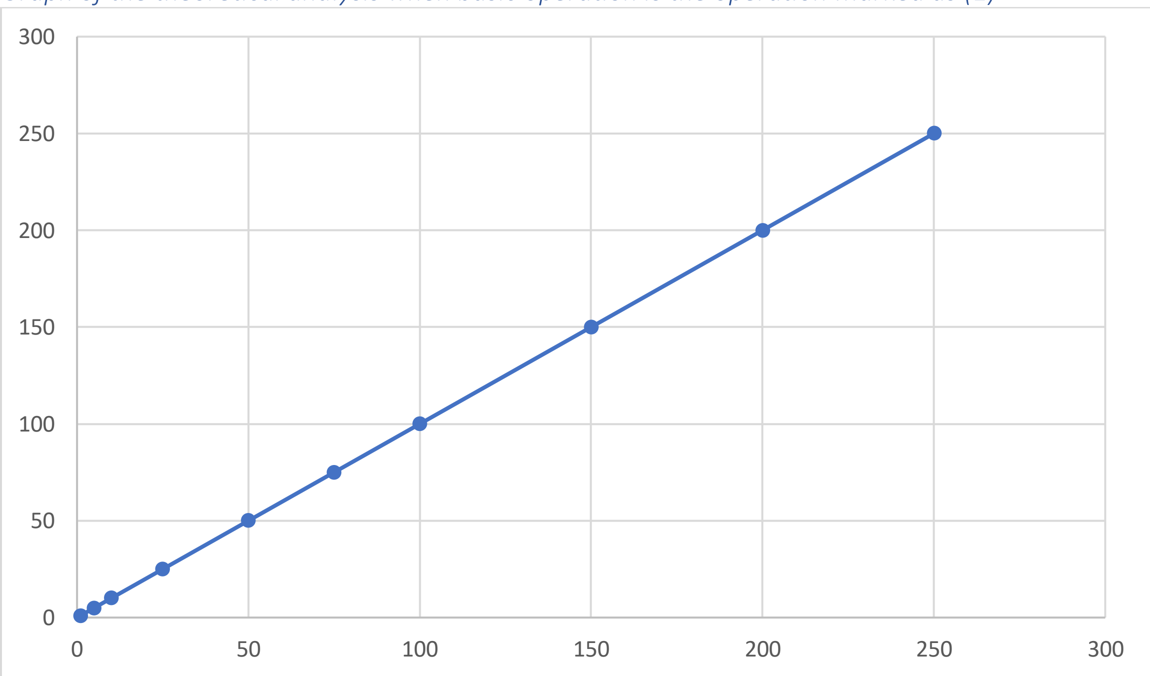
ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

Average Case

Graph of the real execution time of the algorithm

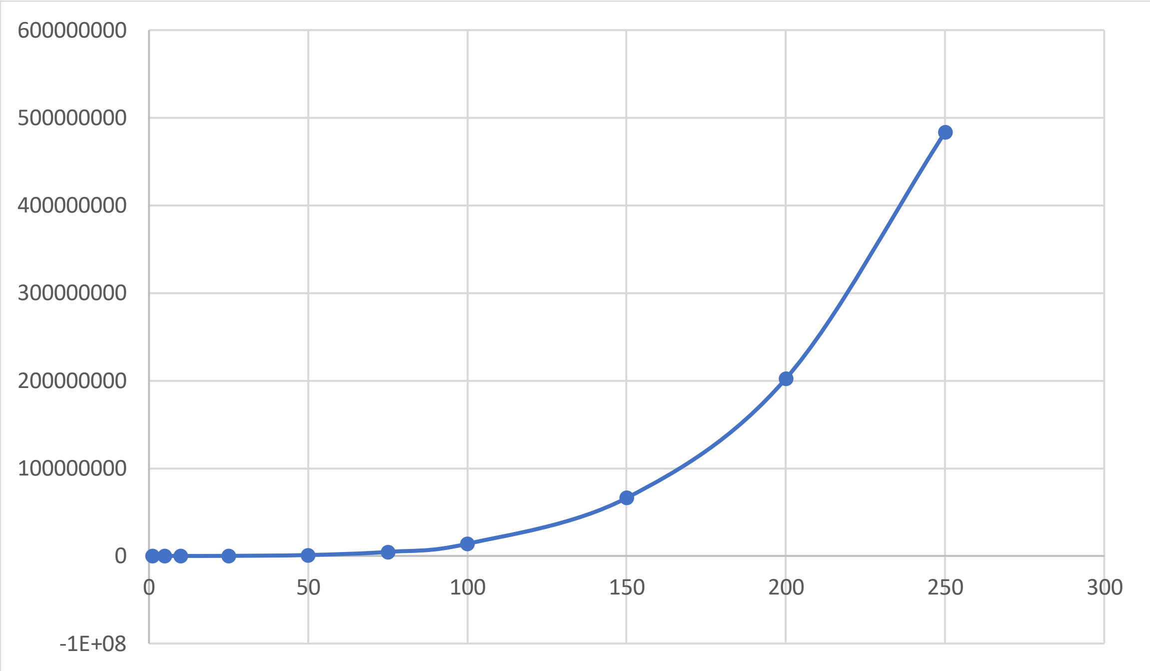


Graph of the theoretical analysis when basic operation is the operation marked as (1)

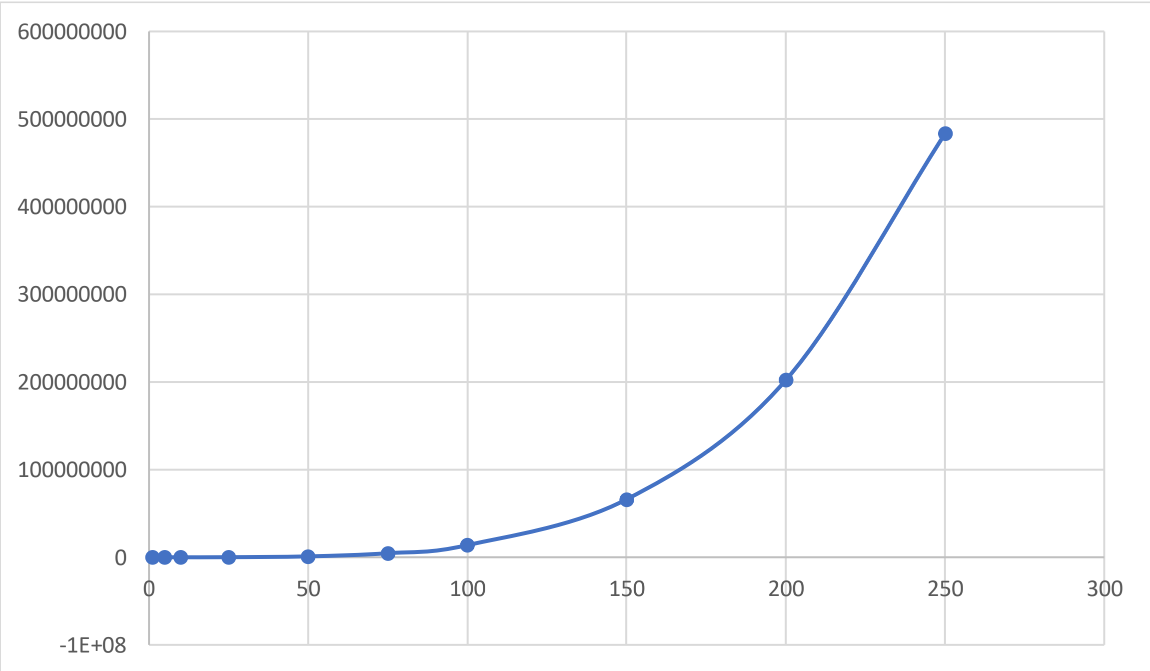


ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

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

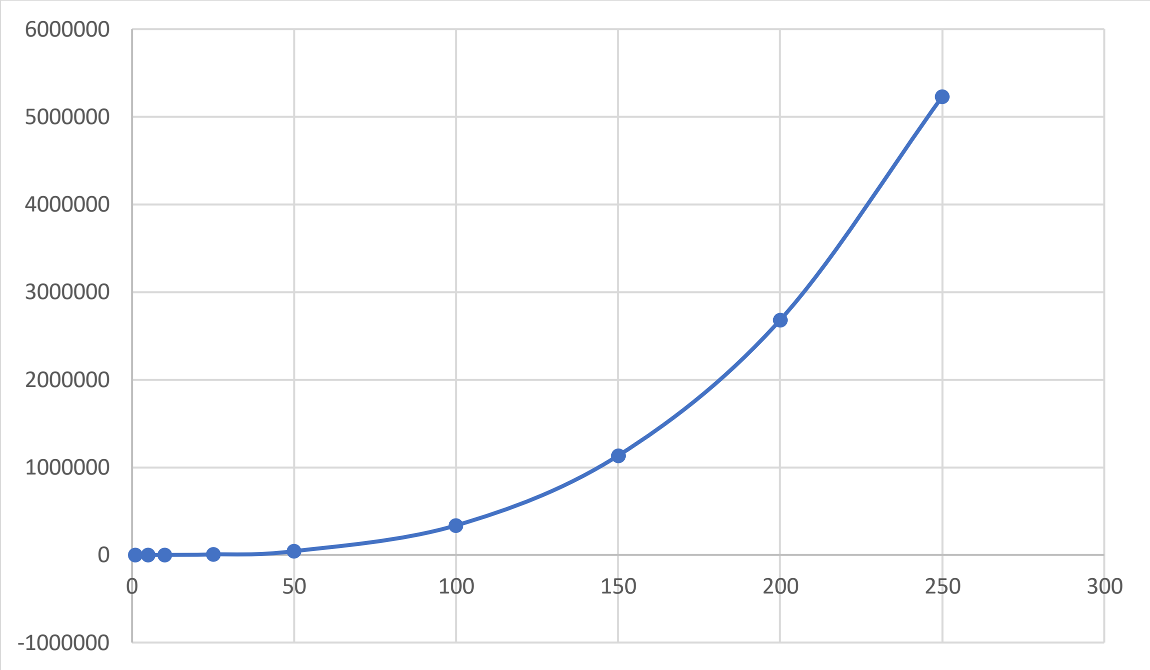


Graph of the theoretical analysis when basic operation is the operation marked as (3)

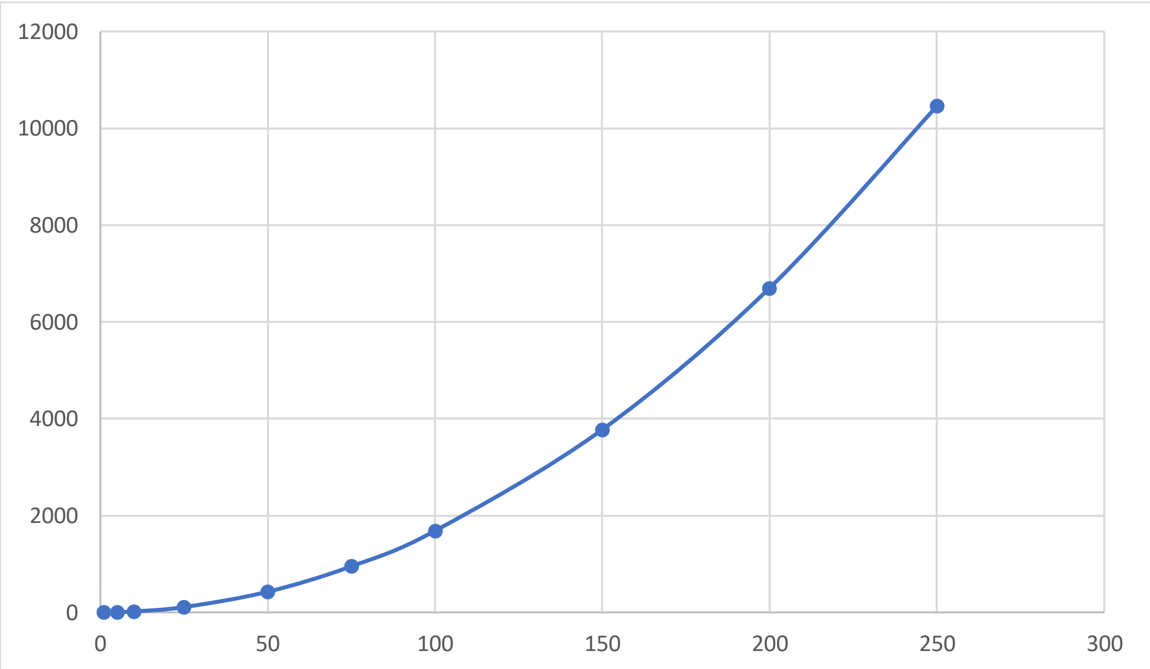


ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

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



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



Comments

Again, the first and third graphs have similar growth rates. This shows that our decision on the basic operation is correct. The fourth graph also has a similarity to these two graphs since its related theoretical analysis has $\Theta(n^4)$ asymptotic notation which is also asymptotic notation of the related theoretical analysis of the third graph but this similarity is not as strong as the previous one. The 2nd, 5th and 6th graphs are growing slower than the real execution time graph since their corresponding complexities are slower than n^4 as the input size goes to infinity.

ALİ BAŞARAN – 2020400357
ÖMER ŞAFAK BEBEK – 2019400180

The growth rates of operation marked as 2's graphs and the real execution graphs are similar in the best case, the worst case and the average case. In conclusion the operation marked as 2 is the most appropriate operation to be a basic operation.