HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

Table of Contents

HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

THEORETICAL ANALYSIS

Basic operation is the comparison marked as (1)

Analyze B(n)

Regardless of the elements in the array, operation (1) should be executed one time in every iteration of the outermost for loop. We know that comparison is done by exactly one basic operation. That means for every input with size n, this algorithm executes operation (1) n times:

$$B(n) = W(n) = \sum_{i=0}^{n-1} 1 = n \Rightarrow B(n) \in \theta(n) \; and \; W(n) \in \theta(n)$$

Analyze W(n)

Regardless of the elements in the array, operation (1) should be executed one time in every iteration of the outermost for loop. We know that comparison is done by exactly one basic operation. That means for every input with size n, this algorithm executes operation (1) n times:

$$B(n) = W(n) = \sum_{i=0}^{n-1} 1 = n \Rightarrow B(n) \in \theta(n) \; and \; W(n) \in \theta(n)$$

Analyze A(n)

We know that $B(n) \le A(n) \le W(n)$. Since the number of operations in both the best and worst case is n, we can safely say that the average number of operations is also n. So,

$$B(n) = n \; and \; W(n) = n \Rightarrow A(n) = n \Rightarrow A(n) \in \theta(n)$$

Basic operations are the two loop incrementations marked as (2)

Analyze B(n)

In every iteration of the outermost for loop, regardless of either X[i]=0 or X[i]=1, the first inner loop always executes. So, loop incrementation always occurs. That means, we can not choose the best and worst input with size n because every input with size n has to do the same number of operations. We know that loop incrementation is 1 basic operation and the first inner loop executes n-i times for each iteration of the outer for loop:

$$B(n) = \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} 1 = \sum_{i=0}^{n-1} n - i = (n) + (n - 1) + (n - 2) + ... + 2 + 1 = \frac{(n+1)\cdot n}{2} = \frac{n^2+n}{2}$$

$$\Rightarrow B(n) \in \theta(n^2) \; and \; W(n) \in \theta(n^2)$$

Analyze W(n)

In every iteration of the outermost for loop, regardless of either X[i]=0 or X[i]=1, the first inner loop always executes. So, loop incrementation always occurs. That means, we can not choose the best and worst input with size n because every input with size n has to do the same number of operations. We know that loop incrementation is 1 basic operation and the first inner loop executes n-i times for each iteration of the outer for loop:

$$B(n) = \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} 1 = \sum_{i=0}^{n-1} n - i = (n) + (n - 1) + (n - 2) + ... + 2 + 1 = \frac{(n+1)\cdot n}{2} = \frac{n^2+n}{2}$$

$$\Rightarrow B(n) \in \theta(n^2) \; and \; W(n) \in \theta(n^2)$$

HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

Analyze A(n)

We know that $B(n) \leq A(n) \leq W(n)$. Since the number of operations in both the best and worst case is $\frac{n^2+n}{2}$, we can safely say that the average number of operations is also $\frac{n^2+n}{2}$. So,

$$B(n) = \frac{n^2+n}{2} \; and \; W(n) = \frac{n^2+n}{2} \Rightarrow A(n) = \frac{n^2+n}{2} \Rightarrow A(n) \in \theta(n^2)$$

Basic operation is the assignment marked as (3)

Analyze B(n)

The best case in this scenario occurs when all the elements in the list are 1 since the basic operation occurs only when X[i]=0 for any i. That means, in such a list, no basic operation occurs: number of basic operations is 0 when all the elements are 1:

$$B(n) = 0 \Rightarrow B(n) \in O(1)$$

Analyze W(n)

The worst case occurs when all the elements in the list are 0 since in that case, the basic operation occurs for every element in the list. The innermost loop which covers operation (3) iterates logn+1 times since initially k=n and we divide k by 2 in every iteration of the innermost loop until we reach n=1.

$$W(n) = \sum_{i=0}^{n-1}\sum_{j=i}^{n-1} logn = \sum_{i=0}^{n-1}(n - i) \cdot (logn + 1) = \frac{(n+1)\cdot n}{2} \cdot (logn + 1)$$

$$\Rightarrow W(n) \in \theta(n^2 \cdot logn)$$

This is valid for inputs $n = 2^k$. Since $\frac{(n+1)\cdot n}{2} \cdot (logn + 1)$ and $n^2 \cdot logn$ are both eventually nondecreasing functions and $n^2 \cdot logn$ is θ-invariant, this result can be generalized to any $n \in \mathbb{N}$.

Analyze A(n)

Let f(i) be the number of operations when the element X[i] at index i is 0, and let g(i) be the number of operations when the element X[i] at index i is 1. Let p_0 be the probability of X[i] being 0, and let p_1 be the probability of X[i] being 1. Then the average number of operations at index i is $f(i) \cdot p_0 + g(i) \cdot p_1$. We know that $p_0 = 1/3, p_1 = 2/3,$

$f(i) = \sum_{j=i}^{n-1} logn + 1$ and $g(i) = 0$. So, the total average number of operations $A(n)$ is:

$$A(n) = \sum_{i=0}^{n-1} f(i) \cdot p_0 + g(i) \cdot p_1$$

$$= \sum_{i=0}^{n-1}\left[\left(\sum_{j=i}^{n-1}(logn + 1) \cdot \frac{1}{3}\right) + 0 \cdot \frac{2}{3}\right]$$

$$= \frac{1}{3}\sum_{i=0}^{n-1}(n - i) \cdot (logn + 1)$$

$$= \frac{1}{3}\frac{(n+1)\cdot n}{2} \cdot (logn + 1)$$

$$\Rightarrow A(n) \in \theta(n^2 \cdot logn)$$

This is valid for inputs $n = 2^k$. Since $\frac{1}{3}\frac{(n+1)\cdot n}{2} \cdot (logn + 1)$ and $n^2 \cdot logn$ are both eventually nondecreasing functions and $n^2 \cdot logn$ is θ-invariant, this result can be generalized to any $n \in \mathbb{N}$.

HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

Basic operations are the two assignments marked as (4)

This time, for every X[i]=0, $\sum_{j=i}^{n-1}(logn + 1)$ operations are performed. On the other side, for every X[i]=1, $\sum_{m=i}^{n-1}\sum_{t=1}^{n}\frac{n}{t}$ operations are performed. We know from our previous discussions in the lectures that harmonic series sum $\sum_{t=1}^{n}\frac{1}{t} \in \theta(logn)$. The number of basic operations for every X[i]=0 is $\sum_{j=i}^{n-1}(logn + 1) = (n - i) \cdot (logn + 1)$ which is $\theta(n \cdot logn)$(This analysis is valid for inputs $n = 2^k$. Since $(n - i) \cdot (logn + 1)$ and $n \cdot logn$ are both eventually nondecreasing functions for a fixed i, and $n \cdot logn$ is θ-invariant, this result can be generalized to any $n \in \mathbb{N}$.). On the other hand, the number of basic operations for every X[i]=1 is

$$\sum_{m=i}^{n-1}\sum_{t=1}^{n}\frac{n}{t} = n \cdot \sum_{m=i}^{n-1}\sum_{t=1}^{n}\frac{1}{t} = n \cdot \sum_{m=i}^{n-1} H(n) = n \cdot H(n) \cdot \sum_{m=i}^{n-1} 1 = n \cdot (n - i) \cdot H(n)$$

which is $\theta(n^2 \cdot logn)$ since $H(n) \in \theta(logn)$. This means for every X[i]=1, more operations are performed than that of the case when X[i]=0 is performed.

Analyze B(n)

The above explanation gives us that the best case occurs when all the elements in the list are 0. In this case number of operations is:

$$B(n) = \sum_{i=0}^{n-1}\sum_{j=i}^{n-1}(logn + 1) = \sum_{i=0}^{n-1}(n - i) \cdot (logn + 1) = \frac{(n+1)\cdot n}{2} \cdot (logn + 1)$$

$$\Rightarrow B(n) \in \theta(n^2 \cdot logn)$$

Analyze W(n)

The above explanation also gives us that the worst case occurs when all the elements in the list are 1. In this case number of basic operations is:

$$W(n) = \sum_{i=0}^{n-1}\sum_{m=i}^{n-1}\sum_{t=1}^{n}\frac{n}{t} = n \cdot \sum_{i=0}^{n-1}\sum_{m=i}^{n-1}\sum_{t=1}^{n}\frac{1}{t} = \sum_{i=0}^{n-1}\sum_{m=i}^{n-1} H(n) = n \cdot H(n) \cdot \sum_{i=0}^{n-1}(n - i)$$
$$= n \cdot H(n) \cdot \frac{(n+1)\cdot n}{2}$$

We know that $H(n) \in \theta(logn)$. So,

$$\Rightarrow W(n) \in \theta(n^3 \cdot logn)$$

Analyze A(n)

Let f(i) be the number of operations when the element X[i] at index i is 0, and let g(i) be the number of operations when the element X[i] at index i is 1. Let p_0 be the probability of X[i] being 0, and let p_1 be the probability of X[i] being 1. Then the number of operations at index i is $f(i) \cdot p_0 + g(i) \cdot p_1$. We know that $p_0 = 1/3, p_1 = 2/3, f(i) = \sum_{j=i}^{n-1}(logn + 1)$ and $g(i) = \sum_{m=i}^{n-1}\sum_{t=1}^{n}\frac{n}{t}$. So, the average number of operations $A(n)$ is:

HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

$$A(n) = \sum_{i=0}^{n-1} f(i) \cdot p_0 + g(i) \cdot p_1$$

$$= \sum_{i=0}^{n-1}\left[\left(\sum_{j=i}^{n-1} (logn + 1) \cdot \frac{1}{3}\right) + \left(\sum_{m=i}^{n-1} \sum_{t=1}^{n} \frac{n}{t} \cdot \frac{2}{3}\right)\right]$$

$$= \sum_{i=0}^{n-1}\left[(n - i) \cdot (logn + 1) \cdot \frac{1}{3}\right] + \left[n \cdot (n - i) \cdot H(n) \cdot \frac{2}{3}\right]$$

$$= \sum_{i=0}^{n-1}\left[(n - i) \cdot (logn + 1) \cdot \frac{1}{3}\right] + \sum_{i=0}^{n-1}\left[n \cdot (n - i) \cdot H(n) \cdot \frac{2}{3}\right]$$

$$= \frac{1}{3} \cdot \frac{(n+1) \cdot n}{2} \cdot (logn + 1) + \frac{2}{3} \cdot n \cdot \frac{(n+1) \cdot n}{2} \cdot H(n)$$

$$\Rightarrow A(n) \in \theta(n^3 \cdot logn)$$

IDENTIFICATION OF BASIC OPERATION(S) ...

Here, state clearly which operation(s) in the algorithm must be the basic operation(s). Also, you should provide a simple explanation about why you have decided on the basic operation you choose. (1-3 sentences)

The basic operations of this algorithm are the two assignments marked as (4). Because these operations contribute most to the execution time, for all the elements in the list these operations always execute independent of the value and they are the typical behaviour of this algorithm.

REAL EXECUTION

Best Case

N Size	Time Elapsed (seconds)
1	5.00000000001e-06
10	2.8e-05
50	0.000701
100	0.002831
200	0.011339
300	0.026239
400	0.05168
500	0.073826
600	0.109212
700	0.162789

Worst Case

N Size	Time Elapsed (seconds)
1	3e-06
10	0.000161
50	0.020314
100	0.142329
200	1.149125
300	4.530324
400	10.8144

500	20.524673
600	38.353826
700	64.525506

Average Case

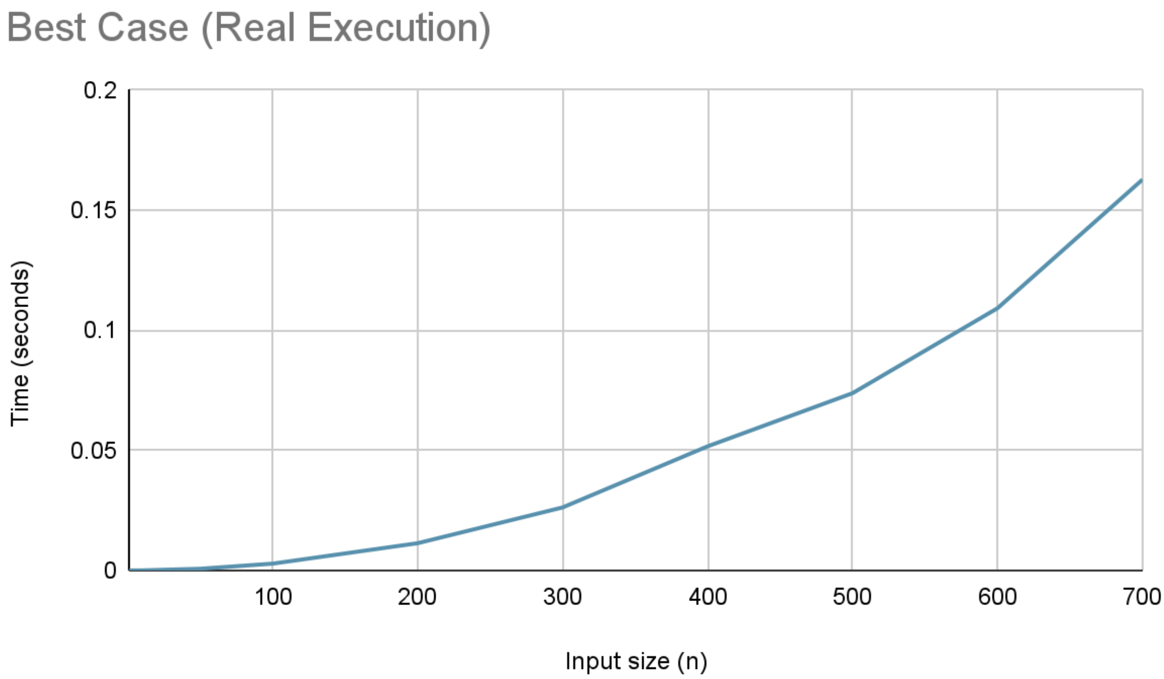
N Size	**Time Elapsed**
1	2.66666666667e-06
10	0.000135
50	00.0117833333333
100	0.0816576666667
200	0.799562333333
300	2.91675
400	6.674457
500	14.7090803333
600	27.1739003333
700	42.8797313333

HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

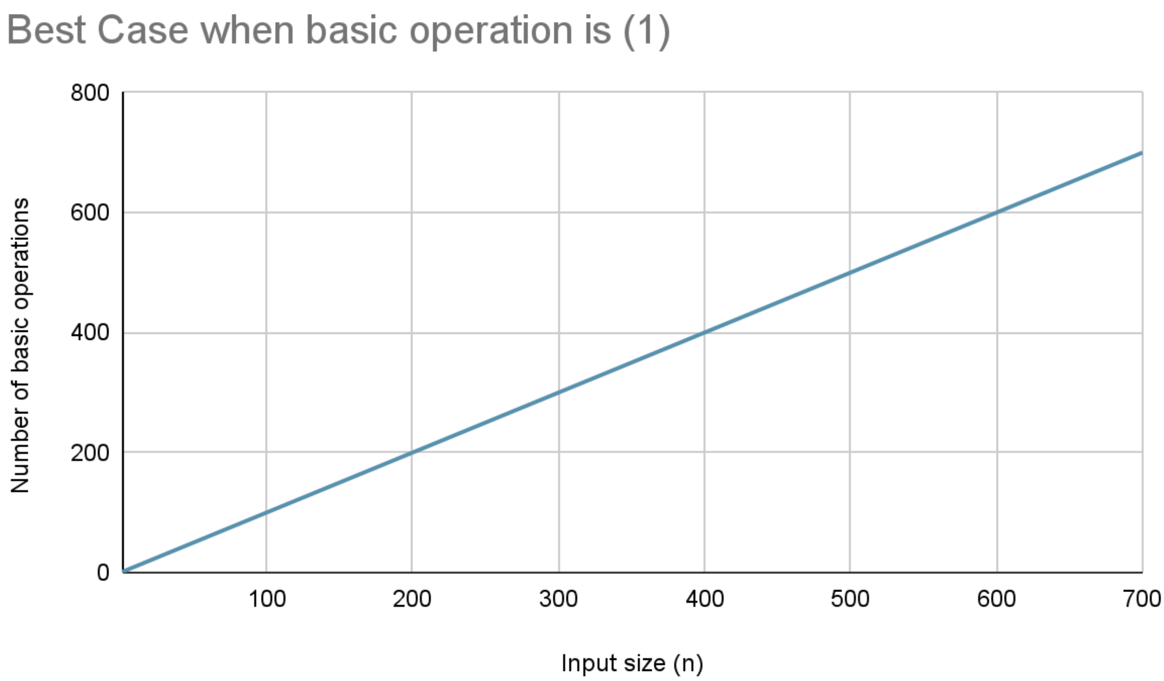
COMPARISON

Best Case

Graph of the real execution time of the algorithm

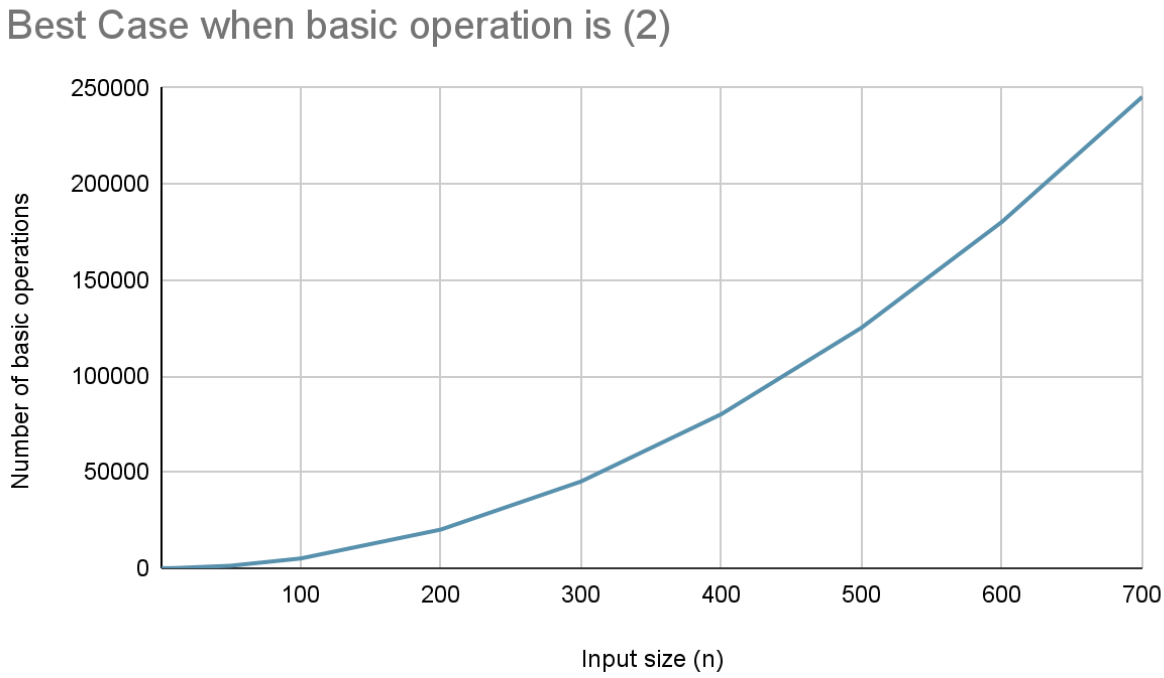


Graph of the theoretical analysis when basic operation is the operation marked as (1)

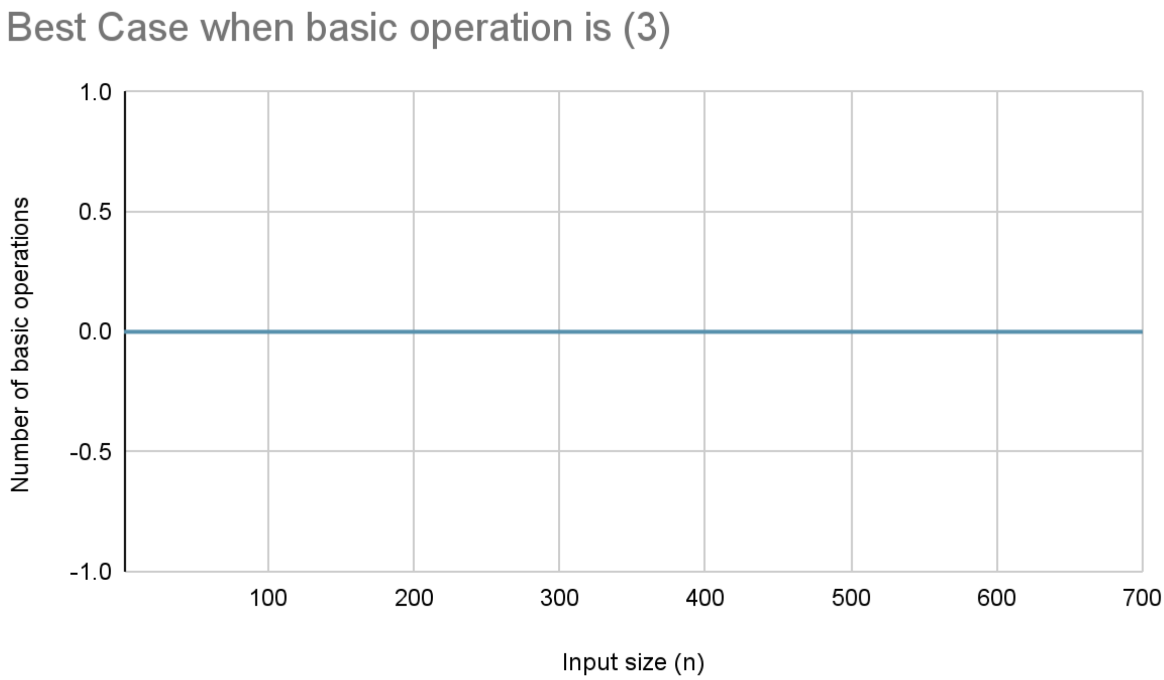


HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

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

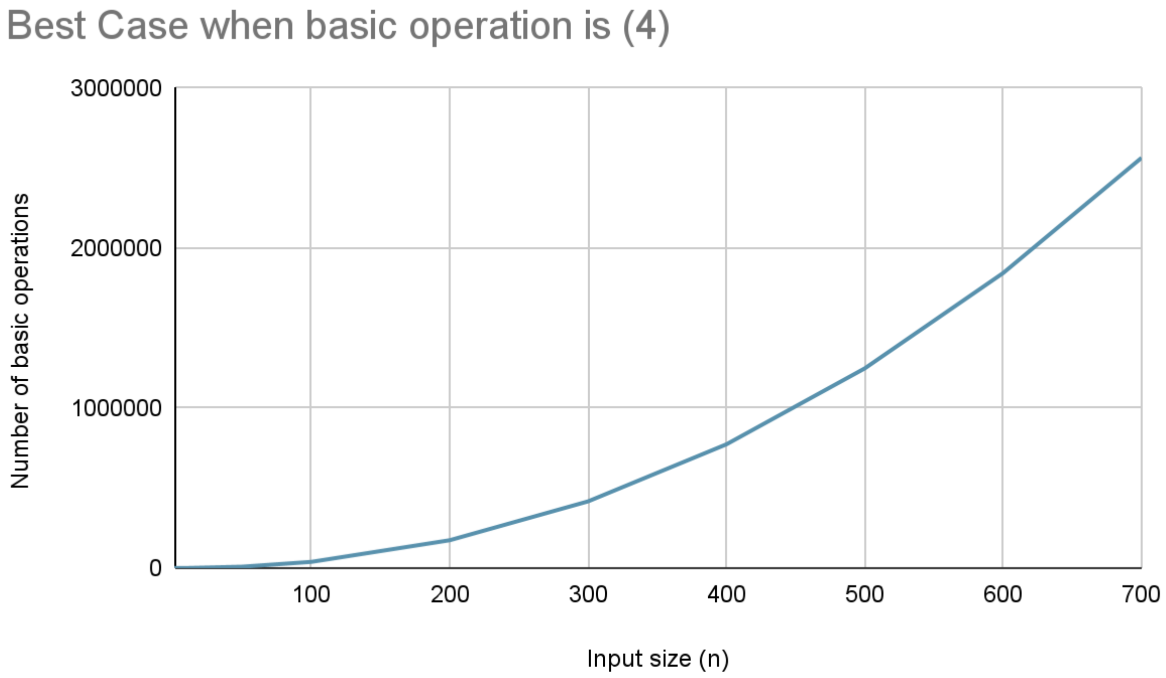


Graph of the theoretical analysis when basic operation is the operation marked as (3)



HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

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



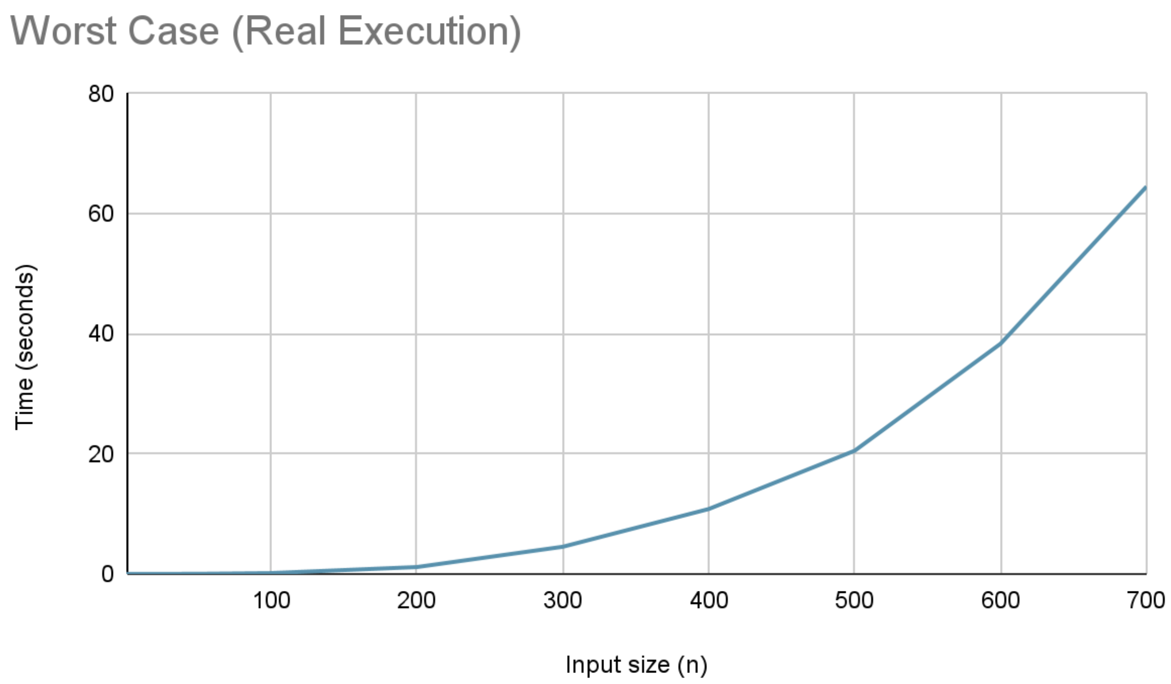
Comments

If we compare these graphs, we can see that the graph of our real analysis is very similar to that of (2) and (4). On the other hand, (1) is not very similar and (3) is not similar at all. In the graph of (3), the number of basic operations does not depend on the input size. We can say that our basic operation may be either (2) or (4) just by looking at the graphs of the best case analysis because the shape of these two operations (2) and (4) are very similar to the shape of the real execution graph. A careful look at the tables can show us that (4) is more similar than (2) to the real case. So, we can say that (4) is more appropriate to be chosen as the realistic basic operation of this algorithm since the real execution graph can be bounded by a constant multiple of the (4)'s graph.

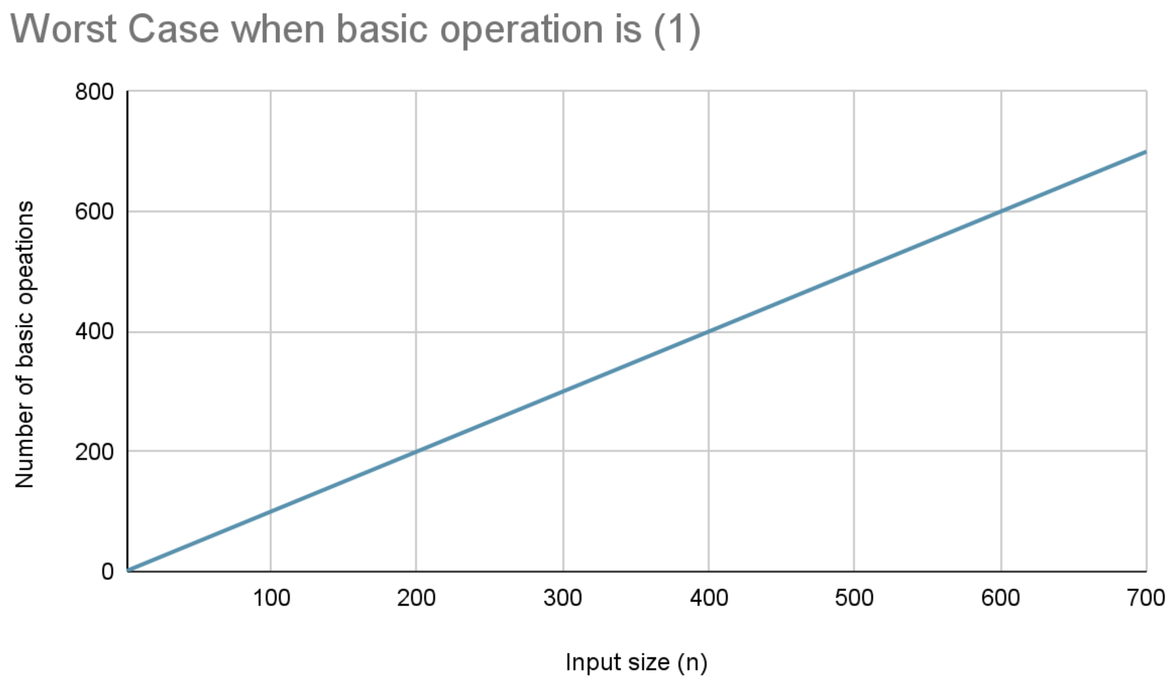
HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

Worst Case

Graph of the real execution time of the algorithm

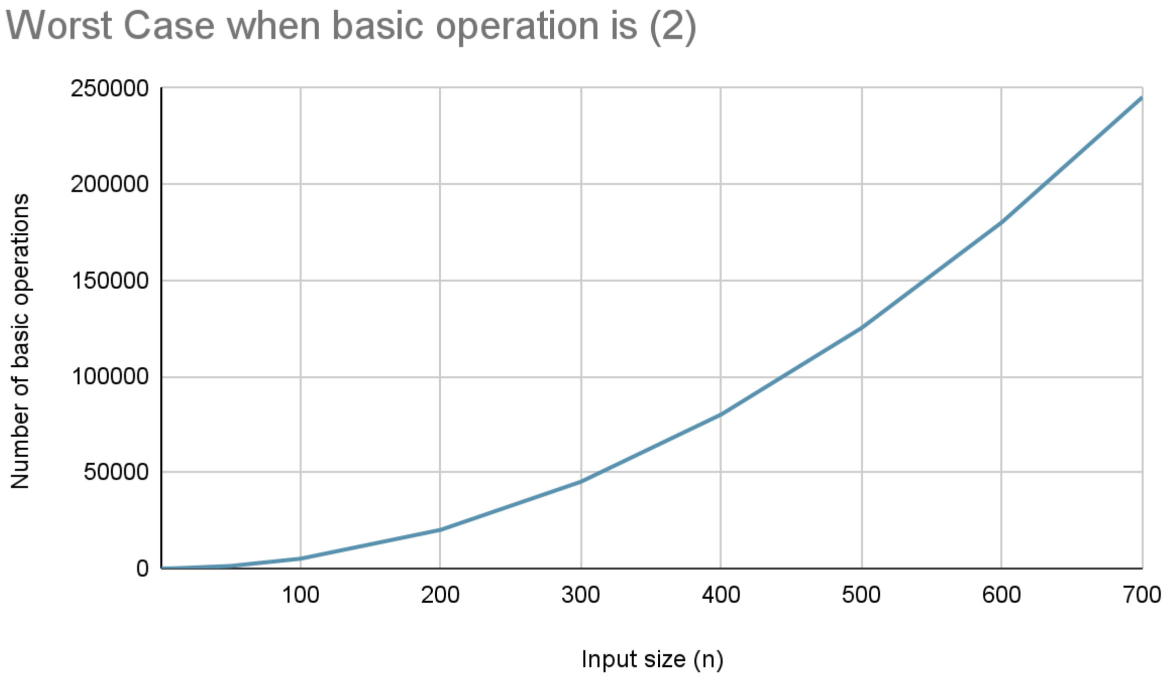


Graph of the theoretical analysis when basic operation is the operation marked as (1)

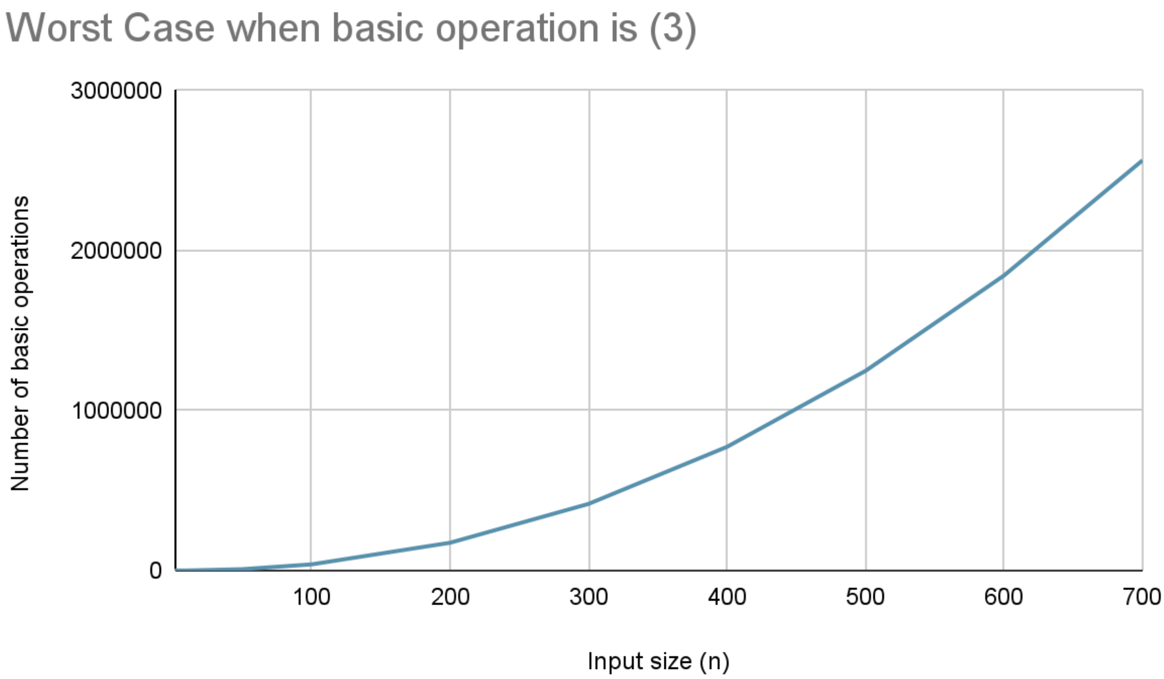


HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

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

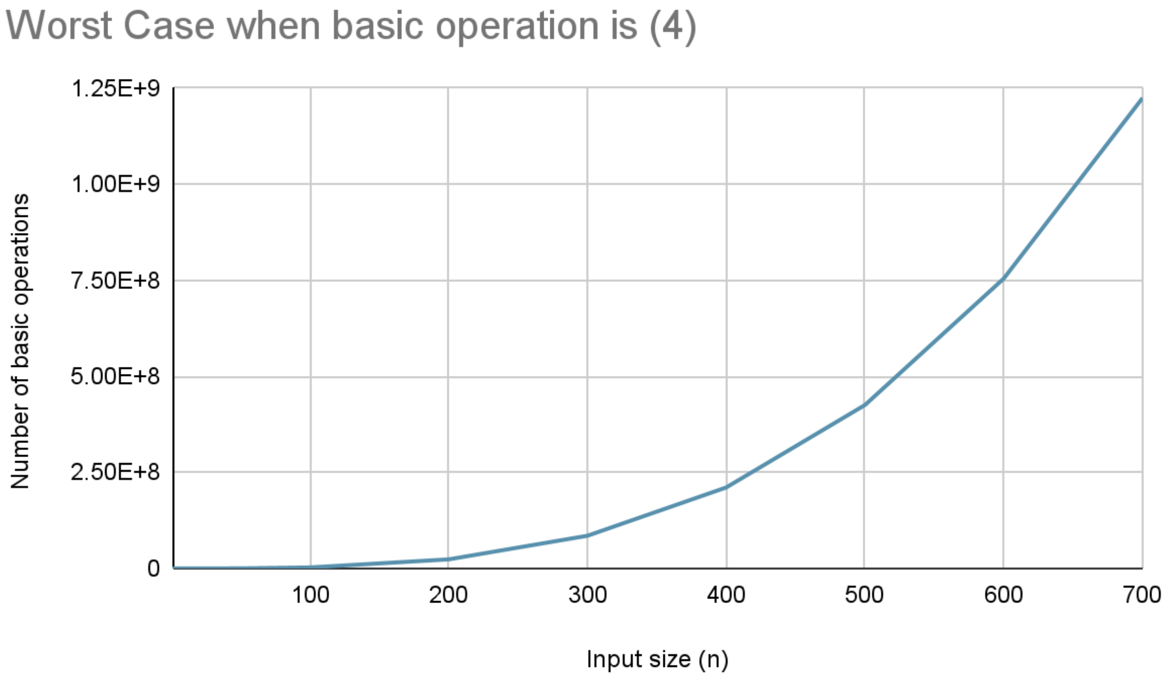


Graph of the theoretical analysis when basic operation is the operation marked as (3)



HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

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



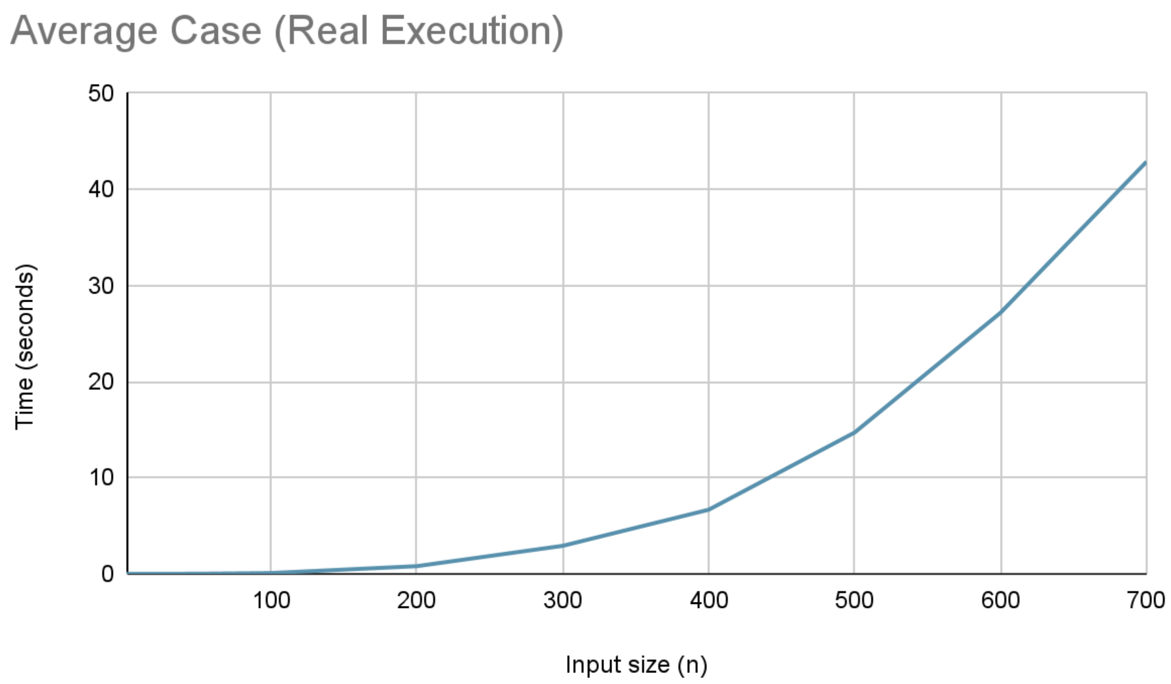
Comments

When we look at the graphs, we can see that the shape of (1) is not similar to the real case. (2), (3) and (4) look similar to the real case. After a careful look, it can be said that (4) is the most similar one to the real case. So, we can say that (4) is the realistic basic operation of this algorithm since the real graph is bounded by constant multiples of (4)'s graph which is not a case for (1), (2) and (3).

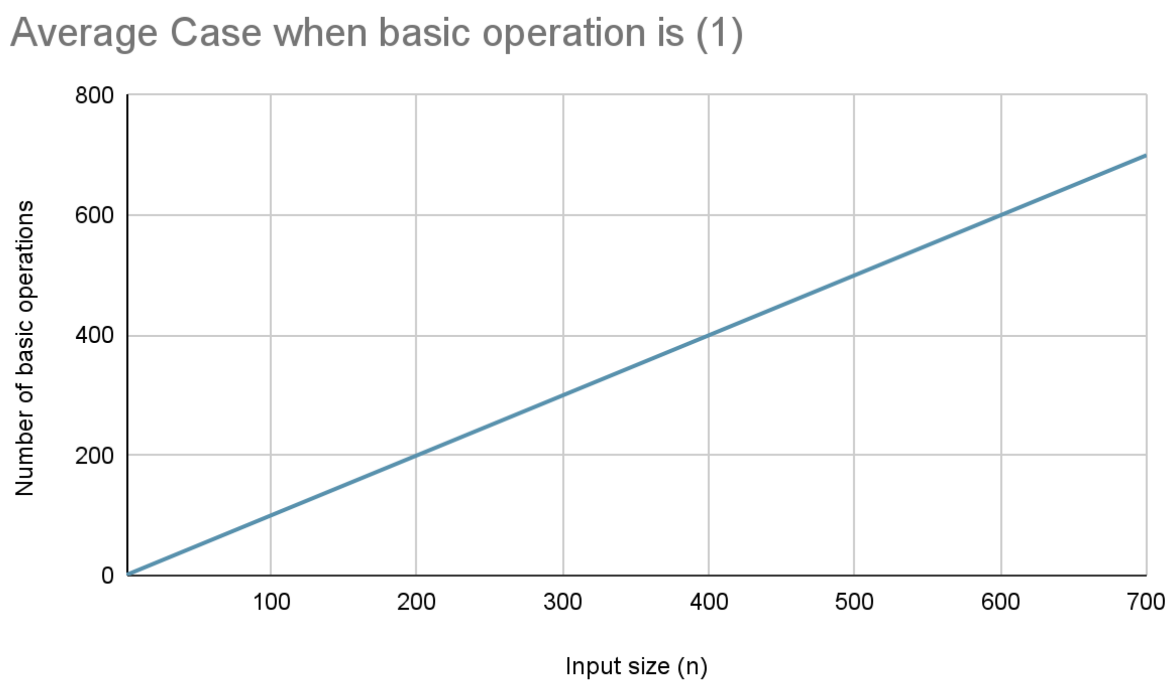
HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

Average Case

Graph of the real execution time of the algorithm

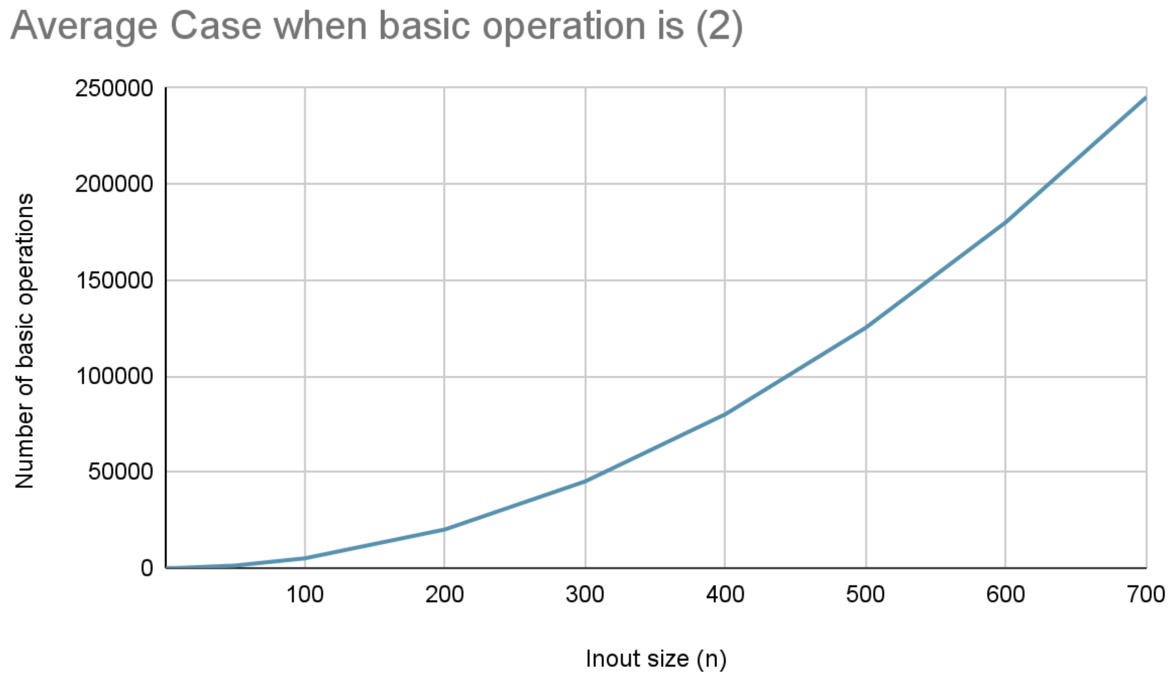


Graph of the theoretical analysis when basic operation is the operation marked as (1)

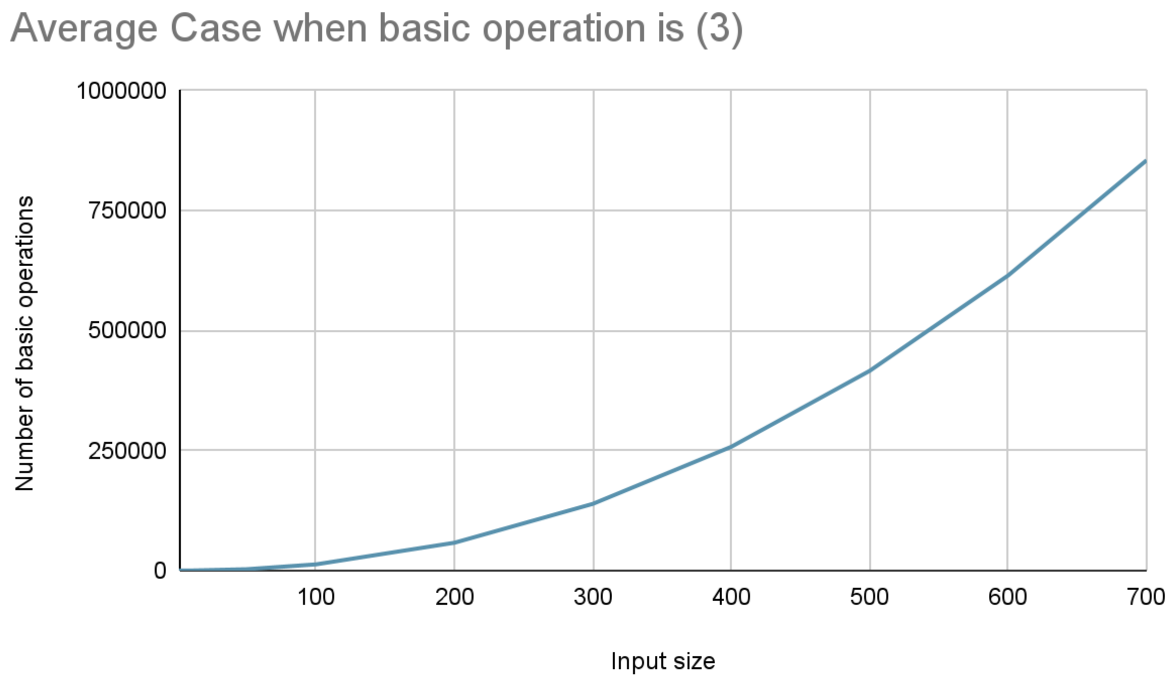


HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

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

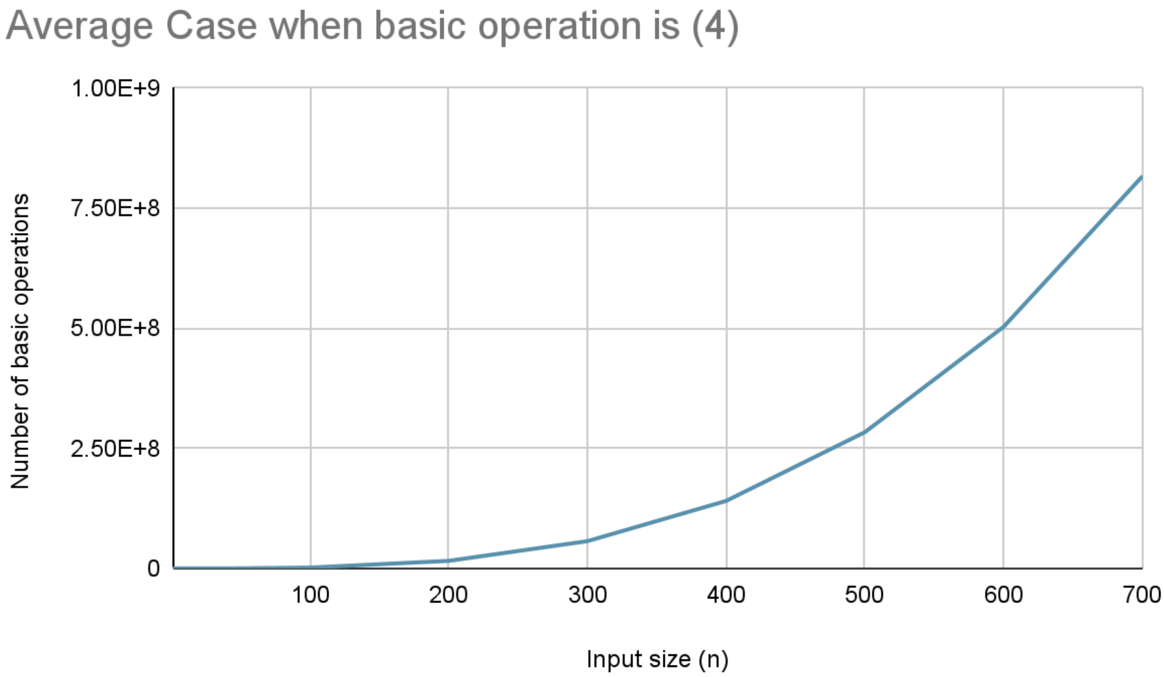


Graph of the theoretical analysis when basic operation is the operation marked as (3)



HALIL BURAK PALA – 2019400282
BERAT DAMAR – 2018400039

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



Comments
When we look at the graphs, we can see that the shape of (1) is not similar to the real case. (2), (3) and (4) look similar to the real case. After a careful look, it can be said that (4) is the most similar one to the real case. So, we can say that (4) is the realistic basic operation of this algorithm since the real graph is bounded by constant multiples of (4)'s graph which is not a case for (1), (2) and (3).