Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

Table of Contents

THEORETICAL ANALYSIS 2
Basic operation is the comparison marked as (1) *2*
Basic operations are the three assignments marked as (2) *2*
Basic operation is two assignments marked as (3) *3*
Basic operations are the two loop incrementations marked as (4) *4*
Basic operation is the assignment marked as (5) *4*
IDENTIFICATION OF BASIC OPERATION(S) 5
REAL EXECUTION 5
Best Case *5*
Worst Case *5*
Average Case *5*
COMPARISON 6
Best Case *6*
Worst Case *8*
Average Case *10*

Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

THEORETICAL ANALYSIS

Basic operation is the comparison marked as (1)

Independent of the input we always perform the operation n times. There is no distinction between best, worst and average cases.

Hence,

$$B(n) = W(n) = A(n) = n \in \theta(n)$$

Analyze B(n)

As stated above,

$$B(n) = n \in \theta(n)$$

Analyze W(n)

As stated above,

$$W(n) = n \in \theta(n)$$

Analyze A(n)

As stated above,

$$A(n) = n \in \theta(n)$$

Basic operations are the three assignments marked as (2)

Consider the number of times the basic operation is performed provided that condition m is true is τ_m.

If the first condition is true $(arr[i] == 0)$ then the for loop executes $(n - i)$ times and the while loop will execute k times, which is $logn$. Therefore, in total there will be $(n - i) * logn$ number of executions of the basic operation.

$$\tau_1 = (n - i) * logn$$

If the second condition is true $(arr[i] == 1)$ then both the outer and the inner for loops execute n times and the while loop will execute $logn$ times as discussed above. Hence,

$$\tau_2 = (n^2) * logn$$

Outer for loop executes n times. Inner for loop will execute $t3^2$ times and values of $t3$ ranges from 1 to n. So the total number of operations will be

$$\tau_3 = \sum_{t3=1}^{n} t3^2 = \frac{n(n+1)(2n+1)}{6}$$

Analyze B(n)

Since τ_1 is the smallest, best case occurs when the first condition is always true.

$$B(n) = \sum_{i=0}^{n-1} \tau_1 = \sum_{i=0}^{n-1} (n-i)logn = logn \sum_{j=1}^{n} j = logn * \frac{n(n+1)}{2}$$

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

Analyze W(n)

Since τ_3 is the greatest, worst case occurs when the third condition is always true.

$$W(n) = \sum_{i=0}^{n-1} \tau_3 = \sum_{i=0}^{n-1} \frac{n(n+1)(2n+1)}{6} = n * \frac{n(n+1)(2n+1)}{6}$$

$$W(n) \in \theta(n^4)$$

Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

Analyze A(n)

The probability of each of the conditions to be true is $\frac{1}{3}$.

$$A(n) = \sum_{i=0}^{n-1} \frac{1}{3}\tau_1 + \frac{1}{3}\tau_2 + \frac{1}{3}\tau_3$$
$$= \frac{1}{3}\sum_{i=0}^{n-1} (n-i) * logn + (n^2) * logn + \frac{n(n+1)(2n+1)}{6}$$
$$= \frac{1}{3}\left(logn * \frac{n(n+1)}{2} + (n^3) * logn + \frac{n^2(n+1)(2n+1)}{6}\right)$$

$$A(n) \in \theta(n^4)$$

Basic operation is two assignments marked as (3)

We will make our analysis similar to the second part.

If the first condition is true $(arr[i] == 0)$ there will be no basic operation execution.

$$\tau_1 = 0$$

If the second condition is true $(arr[i] == 1)$ then both the outer and the inner for loops execute n times and the while loop will execute $logn$ times as discussed above. Hence,

$$\tau_2 = (n^2) * logn$$

Outer for loop executes n times. Inner for loop will execute $t3^2$ times and values of $t3$ ranges from 1 to n. So the total number of operations will be

$$\tau_3 = \sum_{t3=1}^{n} t3^2 = \frac{n(n+1)(2n+1)}{6}$$

Analyze B(n)

Since τ_1 is the smallest, best case occurs when the first condition is always true.

$$B(n) = 0$$
$$B(n) \in \theta(1)$$

Analyze W(n)

Since τ_3 is the greatest, worst case occurs when the third condition is always true.

$$W(n) = \sum_{i=0}^{n-1} \tau_3 = \sum_{i=0}^{n-1} \frac{n(n+1)(2n+1)}{6} = n * \frac{n(n+1)(2n+1)}{6}$$

$$W(n) \in \theta(n^4)$$

Analyze A(n)

$$A(n) = \sum_{i=0}^{n-1} \frac{1}{3}\tau_1 + \frac{1}{3}\tau_2 + \frac{1}{3}\tau_3 = \frac{1}{3}\sum_{i=0}^{n-1} 0 + (n^2) * logn + \frac{n(n+1)(2n+1)}{6}$$
$$= \frac{1}{3}\left(0 + (n^3) * logn + \frac{n^2(n+1)(2n+1)}{6}\right)$$

$$A(n) \in \theta(n^4)$$

Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

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

The basic operations are the 2 loop incrementations. Again the most outer for loop executes n times.

If the second condition is true $(arr[i] == 1)$, the following loop $(for\ t2 \leftarrow n\ downto\ 1\ do)$ executes n times. For each iteration of this loop our basic operation of loop incrementation of $p2$ occurs n times. Hence, there will exactly be n^2 operations.

If the third condition is true $(arr[i] == 2)$, the loop incrementation of $t3$ will occur n times.

Analyze B(n)

The best case occurs when the first condition is always true, which means the basic operations never execute. Hence,

$$B(n) = 0$$
$$B(n) \in \theta(1)$$

Analyze W(n)

It is observed that the worst case happens when the second condition is always true.

$$W(n) = \sum_{i=0}^{n-1} n^2 = n^3$$

$$W(n) \in \theta(n^3)$$

Analyze A(n)

The probability of each of the conditions to be true is $\frac{1}{3}$.

$$A(n) = \sum_{i=0}^{n-1} \frac{1}{3} * 0 + \frac{1}{3} * n^2 + \frac{1}{3} * n = \frac{1}{3} * n * (n^2 + n) = \frac{1}{3}(n^3 + n^2)$$

$$A(n) \in \theta(n^3)$$

Basic operation is the assignment marked as (5)

Assignment marked as 5 occurs only when the first condition is true $(arr[i] == 0)$.

Analyze B(n)

Best case occurs when the first condition is not true and therefore no operation occurs.

$$B(n) = 0$$
$$B(n) \in \theta(1)$$

Analyze W(n)

Worst case is when the first condition is true. In that case, there will be $\sum_{i=0}^{n-1} n - i$ operations. Since the outer for loop executes n times.

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

$$W(n) \in \theta(n^2)$$

Analyze A(n)

The probability of each of the conditions to be true is $\frac{1}{3}$.

$$A(n) = \sum_{i=0}^{n-1} \frac{1}{3} * (n - i) + \frac{1}{3} * 0 + \frac{1}{3} * 0 = \frac{1}{3} * \frac{n(n+1)}{2}$$

$$A(n) \in \theta(n^2)$$

Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

IDENTIFICATION OF BASIC OPERATION(S)

The operations marked as (2) must be the basic operations. Because these operations are the most important, hence characteristic, operations; in the algorithm, these operations are executed repeatedly and they tend to perform longer than most of the other operations since they access array elements.

REAL EXECUTION

Best Case

N Size	Time Elapsed
1	0.000002
5	0.000006
10	0.000017
25	0.000115
50	0.000499
75	0.001352
100	0.002266
150	0.005961
200	0.009837
250	0.015278

Worst Case

N Size	Time Elapsed
1	0.000002
5	0.000019
10	0.000189
25	0.006147
50	0.091153
75	0.456626
100	1.427150
150	7.298703
200	22.792678
250	56.791562

Average Case

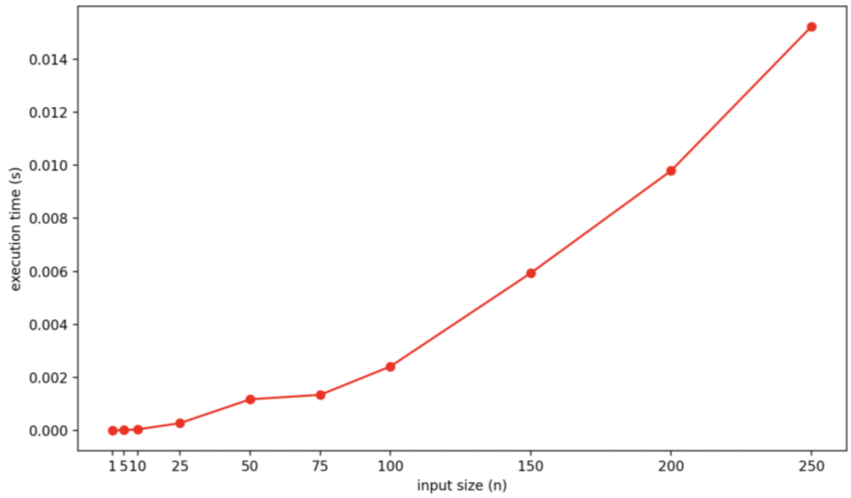
N Size	Time Elapsed
1	0.000003
5	0.000034
10	0.000285
25	0.004577
50	0.031439
75	0.145757
100	0.456666
150	2.194115
200	6.461470
250	14.111794

Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

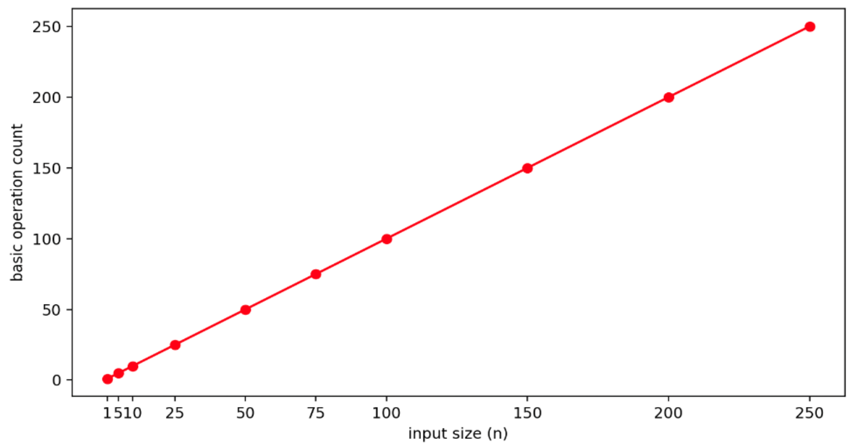
COMPARISON

Best Case

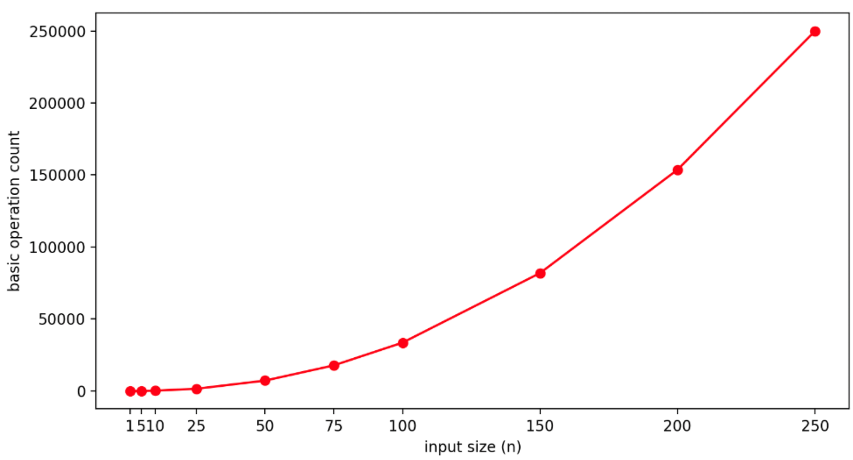
Graph of the real execution time of the algorithm



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

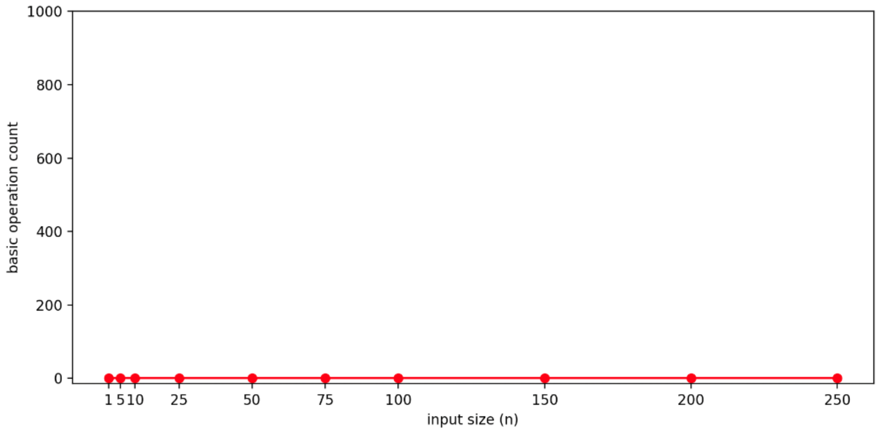


Graph of the theoretical analysis when basic operation is the operation marked as (2)

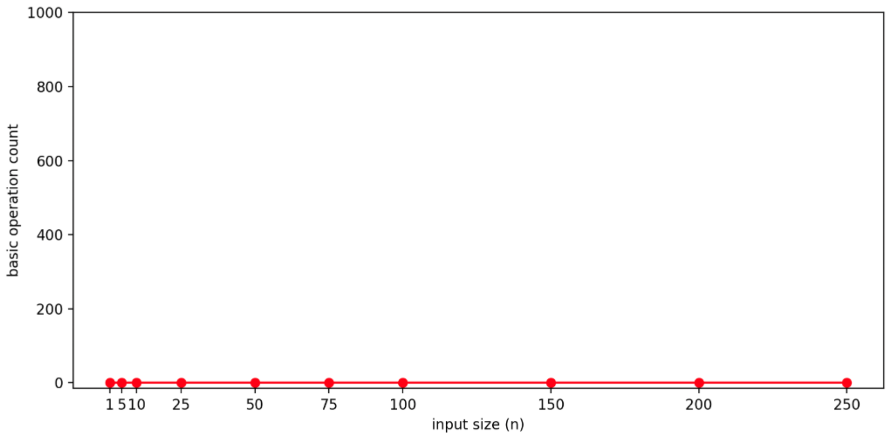


Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

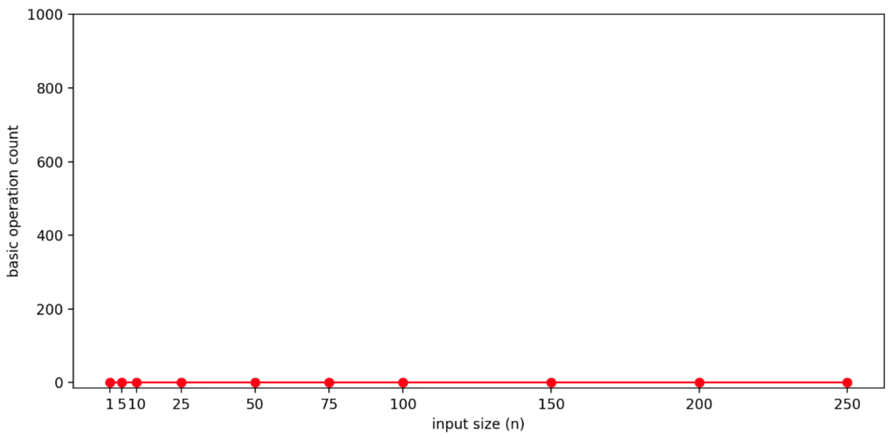
Graph of the theoretical analysis when basic operation is the operation marked as (3)



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)



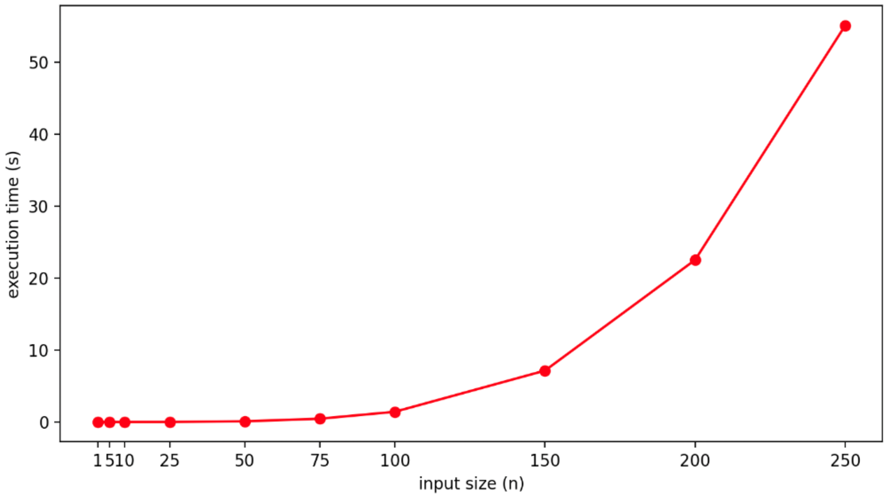
Comments

The graphs when basic operation is the operation (3), (4) and (5) are all constant and 0 since there isn't any execution of the basic operation. When the basic operation is (1), the graph is linear. The growth rate that is the most similar to the growth rate of the real execution time graph is observed in the graph where the basic operation is operation (2) with growth rate $n^2 logn$. This similarity supports our claim that the real basic operation of the algorithm is operation (2).

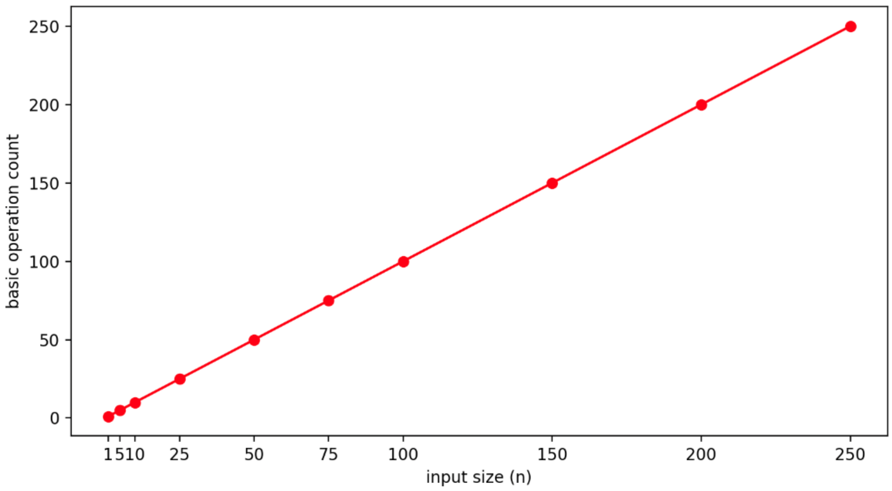
Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

Worst Case

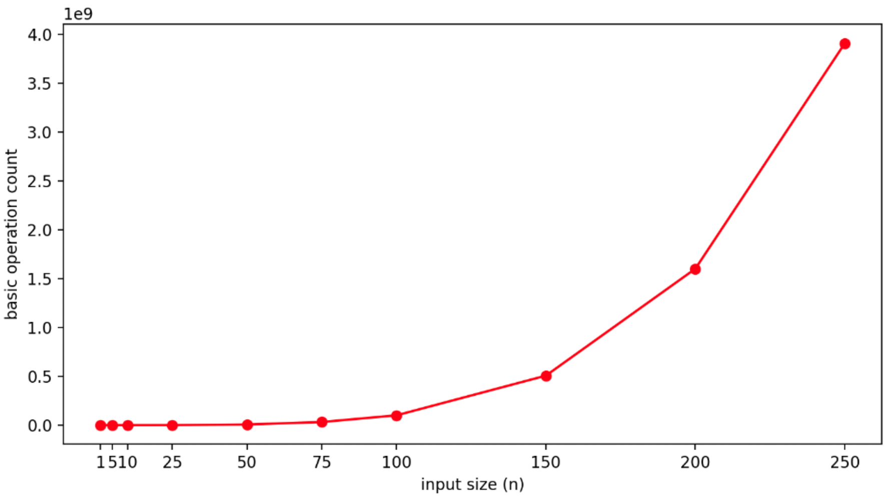
Graph of the real execution time of the algorithm



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

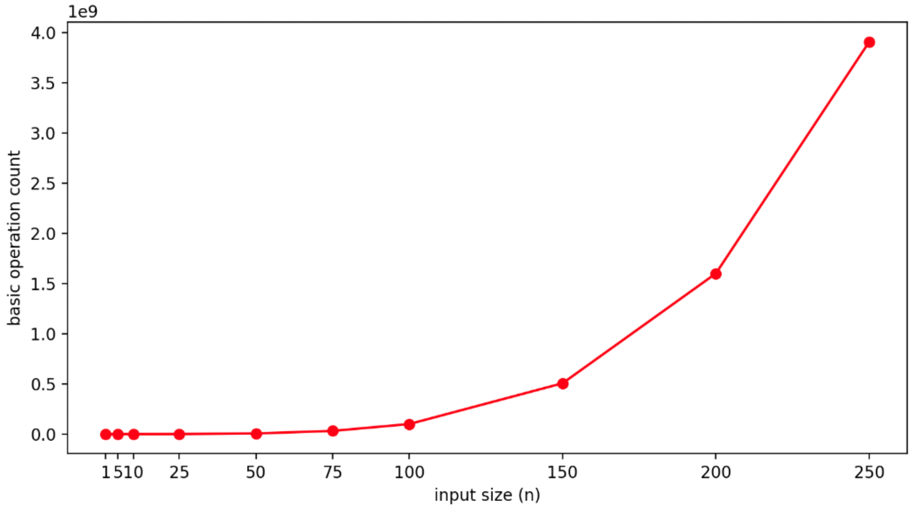


Graph of the theoretical analysis when basic operation is the operation marked as (2)

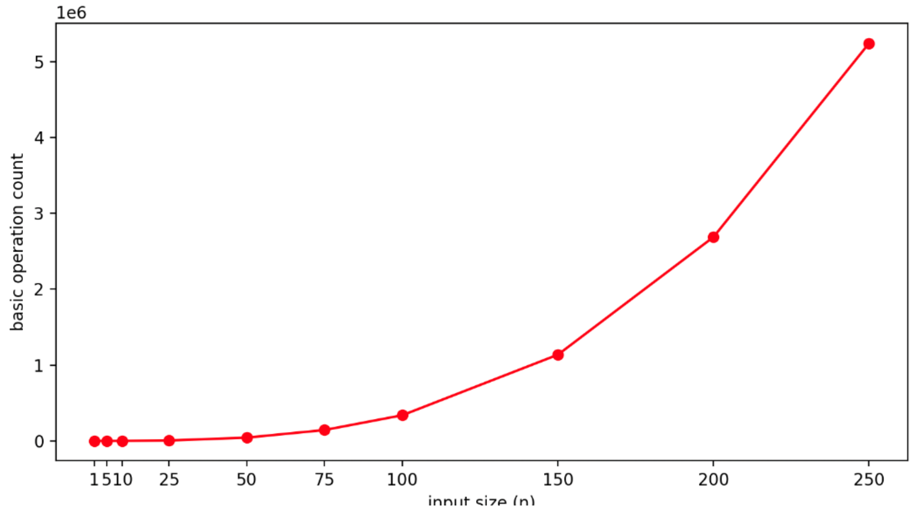


Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

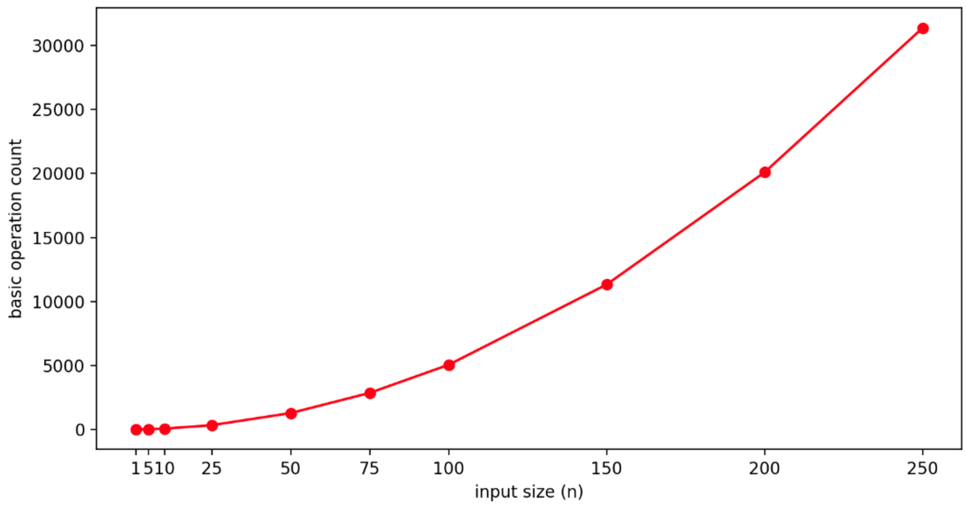
Graph of the theoretical analysis when basic operation is the operation marked as (3)



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)



Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

Comments

Like the analysis for the worst cases, the graph of basic operation (1) is linear whereas (2), (3) are the same and quartic, (4) is cubic and (5) is quadratic. The shapes look similar -except where the basic operation is (1)-, however we can compare the ratios of the execution time and basic operation count as follows:

Ratio of the real execution times when input size is 250 and 200 is:

$$\frac{56.791562}{22.792678} = 2.4916$$

Ratio of number of basic operations for input sizes 250 and 200 when the basic operation is (2) or (3) is:

$$\frac{A(250)}{A(200)} = 2.4224$$

(Even though the exact analysis of (2) and (3) are not the same, they differ with a very small term. Hence the calculated ratios are so close that they can be considered equal.)

Ratio of number of basic operations for input sizes 250 and 200 when the basic operation is (4) is:

$$\frac{(250^3 + 250^2)/3}{(200^3 + 200^2)/3} = 1.9512$$

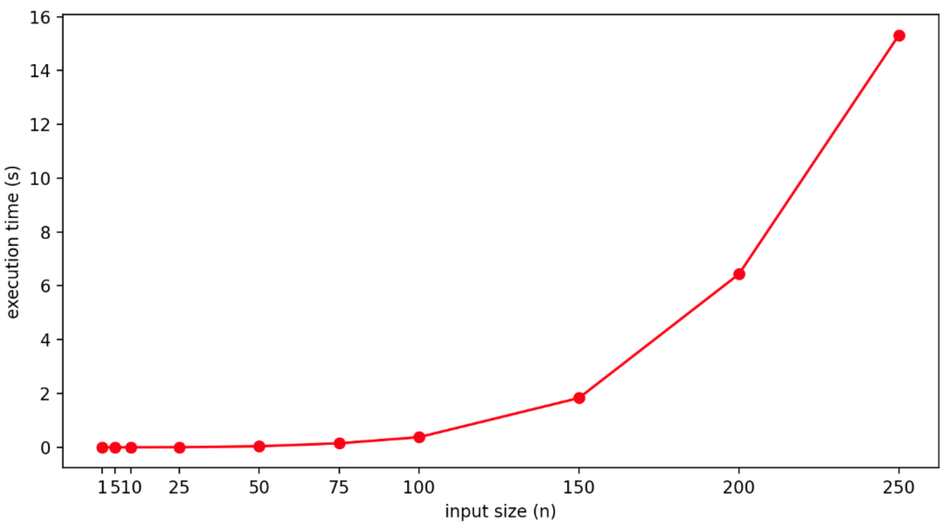
Ratio of number of basic operations for input sizes 250 and 200 when the basic operation is (5) is:

$$\frac{250 * 251/6}{200 * 201/6} = 1.5610$$

By comparing these ratios, it is clearly seen that the growth rate of the algorithm is most similar to the growth rate of the graph where the basic operation is (2) or (3) which is n^4. Again this observation supports our claim that the real basic operation is (2).

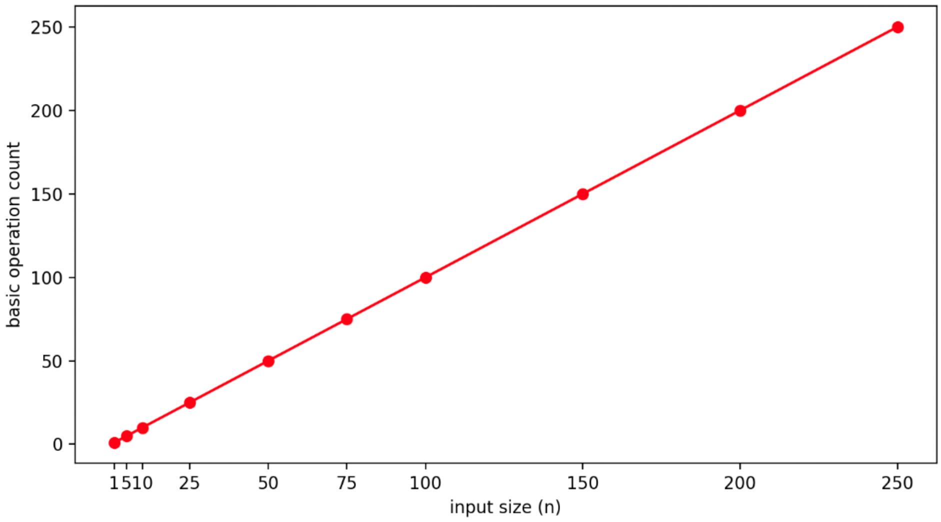
Average Case

Graph of the real execution time of the algorithm

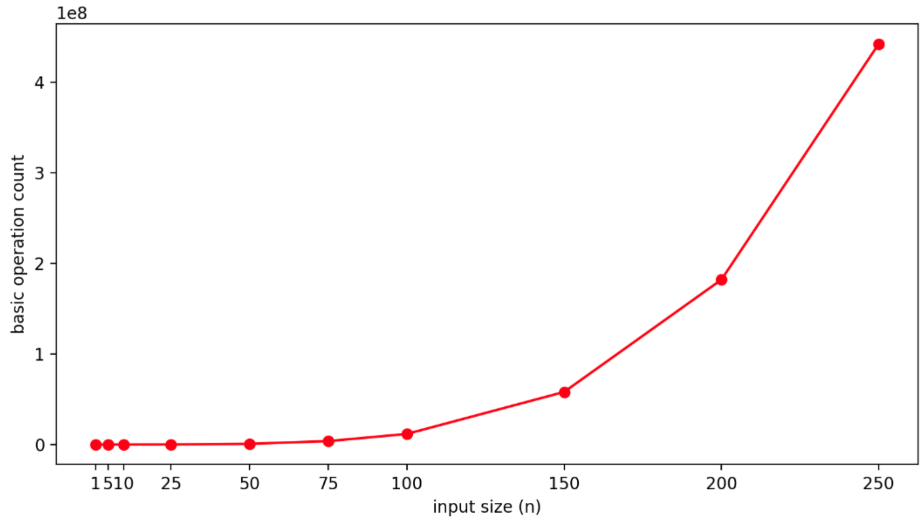


Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

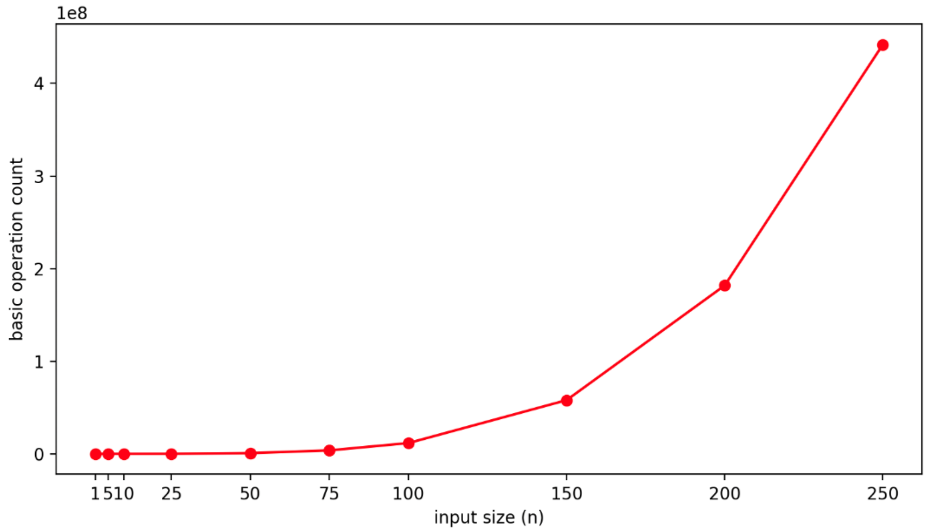
Graph of the theoretical analysis when basic operation is the operation marked as (1)



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

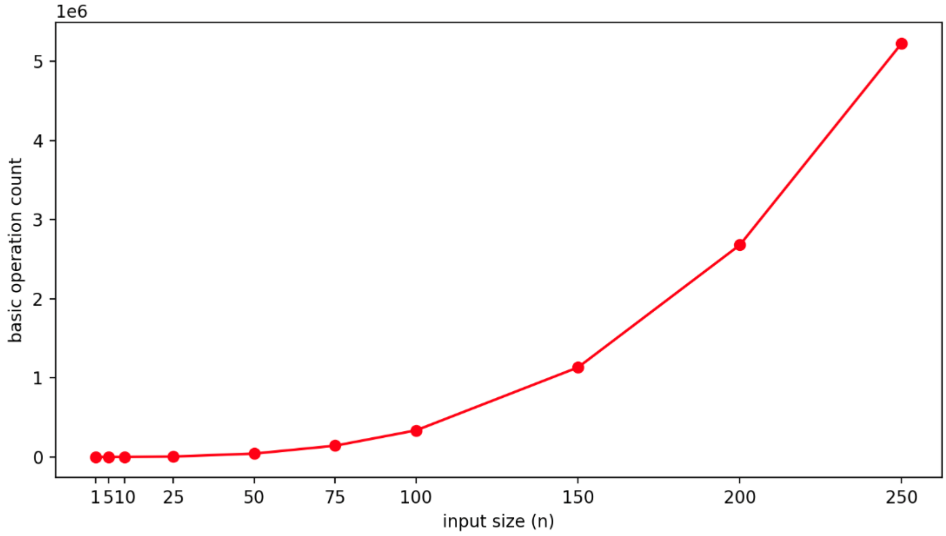


Graph of the theoretical analysis when basic operation is the operation marked as (3)

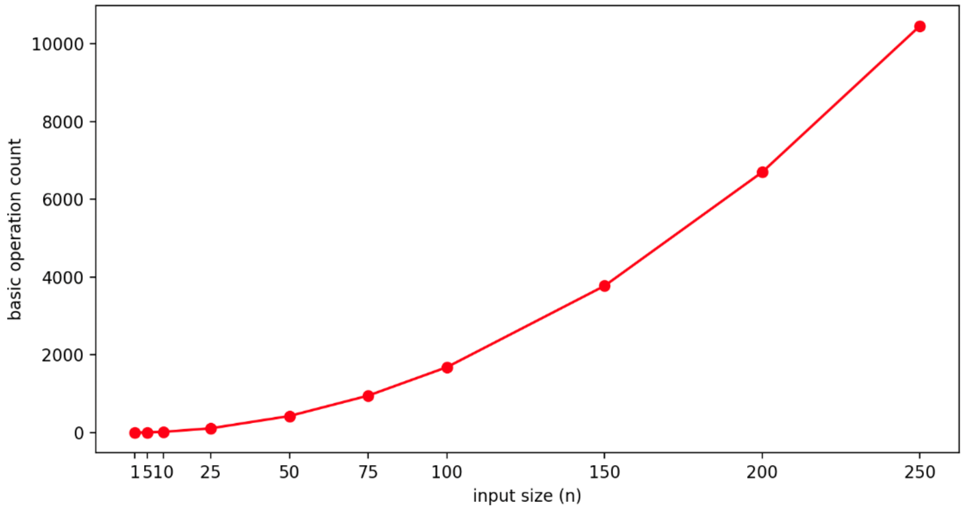


Zeynep Buse Aydın – 2019400066
Furkan Bülbül – 2020400118

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

Similar to the worst case, the graph of the average case when the basic operation is (1) is linear whereas (2), (3) are the same and quartic, (4) is cubic and (5) is quadratic. Since the shapes look similar, we will again compare the ratios:

Ratio of the real execution times when input size is 250 and 200 is:

$$\frac{14.111794}{6.461470} = 2.183991259$$

Ratio of number of basic operations for input sizes 250 and 200 when the basic operation is (2) or (3) is:

$$\frac{250^2(251)(501)/6}{200^2(201)(401)/6} = 2.4378$$

Ratio of number of basic operations for input sizes 250 and 200 when the basic operation is (4) is:

$$\frac{250^3}{200^3} = 1.9531$$

Ratio of number of basic operations for input sizes 250 and 200 when the basic operation is (5) is:

$$\frac{250 * 251}{200 * 201} = 1.5610$$

With this observation, it is clear that the growth rate of the algorithm is most similar to the growth rate of the graph where the basic operation is (2) or (3) which is n^4.