CMPE 300: Project 1

Ahmet Ertuğrul Hacıoğlu
2020400132

Ceyhun Sonyürek
2020400258

November 24, 2023

Contents

1 Theoretical Analysis

1.1 Basic operation is the comparison marked as (1)

1.1.1 Analysis of B(n)

$\rightarrow$ This comparison operation is not dependent of elements of the array. For each value in the array, if it is 0,1, or 2, we have to compare if it is equal to 0 or not.

$$\begin{aligned} B(n) &= \sum_{i=0}^{n-1} 1 \\ &= n \\ &= \Theta(n) \end{aligned}$$

1.1.2 Analysis of W(n)

$$\begin{aligned} W(n) &= \sum_{i=0}^{n-1} 1 \\ &= n \\ &= \Theta(n) \end{aligned}$$

1.1.3 Analysis of A(n)

$$\begin{aligned} A(n) &= \sum_{i=0}^{n-1} 1 \\ &= n \\ &= \Theta(n) \end{aligned}$$

1.2 Basic operations are the two assignments marked as (2)

$\rightarrow$ These assignments can only occur whenever value of the current element in the array at the index i is 0 or 1. Values of elements in the array will affect analysis.

1.2.1 Analysis of B(n)

$\rightarrow$ We can take an array that only consists of *two*s, which result in not executing the given lines because code will continue with the *else* block.

$$\begin{aligned} B(n) &= \sum_{i=0}^{n-1} 0 \\ &= 0 \\ &= O(1) \end{aligned}$$

1.2.2 Analysis of W(n)

→ We can consider would be an array that has no *two*s in it, to increase the code entering *if* or *else if* block. Both *if* and *else if* blocks iterates same number of assignments for both cases. Therefore, let's consider the cases where all of the elements in the array has value of 1.

$$
\begin{aligned}
W(n) &= \sum_{i=0}^{n-1} \sum_{m=i}^{n-1} 1 \\
&\sum_{m=i}^{n-1} 1 = n - i \\
&= \sum_{i=0}^{n-1} n - i \\
&= \sum_{i=0}^{n-1} n - \sum_{i=0}^{n-1} i \\
&= n^2 - \frac{n(n-1)}{2} = \frac{n^2+n}{2} \\
&= \Theta(n^2)
\end{aligned}
$$

1.2.3 Analysis of A(n)

→ In average, $\frac{1}{3}$ of the elements in the array will be $0,\frac{1}{3}$ of the elements in the array will be $1,\frac{1}{3}$ of the elements in the array will be 2. Probabilities of arr[i] $=$ 0, arr[i] $=$ 1 and arr[i] $=$ 2 are $\frac{1}{3}$ since all are equally likely to occur.

$$
\begin{aligned}
A\,(n) &= (1/3) \cdot \sum_{i=0}^{n-1}\sum_{j=i}^{n-1} 1 + (1/3) \cdot \sum_{i=0}^{n-1}\sum_{m=i}^{n-1} 1 + (1/3) \cdot \sum_{i=0}^{n-1} 0 \\
&= (2/3) \cdot \sum_{i=0}^{n-1}\sum_{j=i}^{n-1} 1 \\
\sum_{j=i}^{n-1} &= n - i \\
&= (2/3) \cdot \sum_{i=0}^{n-1} n - i \\
&= (2/3) \cdot (\sum_{i=0}^{n-1} n - \sum_{i=0}^{n-1} i) \\
&= (2/3) \cdot [(n \cdot n) - \frac{n \cdot (n-1)}{2}] \\
&= (2/3) \cdot \frac{n^2 + n}{2} \\
&= \frac{n^2 + n}{3} \\
&= \Theta(n^2)
\end{aligned}
$$

1.3 Basic operations are the two comparisons marked as (3)

→ Comparisons marked as (3) can be executed whenever $arr[i] = 0$ or $arr[i] = 2$ However, in this case they have different number of execution depending on 1 and 2. Therefore, we need to calculate number of execution separately and then sum.

1.3.1 Analysis of B(n)

→ If all the elements in the array have the value 1, then we get the best case because code never executes these given operations.

$$
\begin{aligned}
B\,(n) &= \sum_{i=0}^{n-1} 0 \\
&= O(1)
\end{aligned}
$$

1.3.2 Analysis of W(n)

→ To find worst case elements of array can be 0 or 2 but there is different number of execution for different values of i.Let's compare the number of execution when $arr[i]$ is 0 or 2 to find the maximum of number of execution.We should find the i value that makes both side equal.After that, in worst case, array elements should be 0 until the array[i].

$$\begin{aligned} W(n) &= \sum_{i=0}^{n-1}(\max\left\{\sum_{j=i}^{n-1}\sum_{k=0}^{\lfloor\log_3 n\rfloor+2} 1, \sum_{p=0}^{n} 1\right\}) \\ &= \sum_{i=0}^{n-1}\max\left\{(n-i)\left(\lfloor\log_3 n\rfloor+2\right), n+1\right\} \end{aligned}$$

$$\begin{aligned} (n-i_e)\left(\lfloor\log_3 n\rfloor+2\right) &= n+1 \\ i_e &= n-\left\lfloor\frac{n+1}{\lfloor\log_3 n\rfloor+2}\right\rfloor \\ i_e &= \left\lfloor\frac{n\cdot\lfloor\log_3 n\rfloor+n-1}{\lfloor\log_3 n\rfloor+2}\right\rfloor \end{aligned}$$

$$\begin{aligned} W(n) &= \sum_{i=0}^{i_e}(n-i)(\lfloor\log_3 n\rfloor+2)+\sum_{i=i_e+1}^{n-1} n+1 \\ &= (i_e+1)n\left(\lfloor\log_3 n\rfloor+2\right)-\frac{i_e(i_e+1)\left(\lfloor\log_3 n\rfloor+2\right)}{2}+(n-i_e-1)(n+1) \\ W(n) &= \left(n-\left\lfloor\frac{n+1}{\lfloor\log_3 n\rfloor+2}\right\rfloor+1\right)n\left(\lfloor\log_3 n\rfloor+2\right) \\ &\quad -\frac{\left(n-\left\lfloor\frac{n+1}{\lfloor\log_3 n\rfloor+2}\right\rfloor\right)\left(\left(n-\left\lfloor\frac{n+1}{\lfloor\log_3 n\rfloor+2}\right\rfloor\right)+1\right)\left(\lfloor\log_3 n\rfloor+2\right)}{2} \\ &\quad +\left(n-\left(n-\left\lfloor\frac{n+1}{\lfloor\log_3 n\rfloor+2}\right\rfloor\right)-1\right)(n+1) \\ W(n) &= \frac{3(n^2\lfloor\log_3 n\rfloor)}{2}+3(n^2)-2n\lfloor\log_3 n\rfloor\lfloor\frac{n+1}{\lfloor\log_3 n\rfloor+2}\rfloor \\ &\quad -3\lfloor\frac{n+1}{\lfloor\log_3 n\rfloor+2}\rfloor n+\frac{3}{2}\log_3 n+\frac{(\lfloor\frac{n+1}{\lfloor\log_3 n\rfloor+2}\rfloor^2\lfloor\log_3 n\rfloor)}{2} \\ &\quad +2n+\lfloor\frac{n+1}{\lfloor\log_3 n\rfloor+2}\rfloor^2-\frac{\lfloor\log_3 n\rfloor\lfloor\frac{n+1}{\lfloor\log_3 n\rfloor+2}\rfloor}{2}-1 \\ W(n) &= \Theta(n^2\cdot\log n) \end{aligned}$$

1.3.3 Analysis of A(n)

→ In average, $\frac{1}{3}$ of the elements in the array will be 0,$\frac{1}{3}$ of the elements in the array will be 1,$\frac{1}{3}$ of the elements in the array will be 2. Probabilities of arr[i] = 0, arr[i] = 1 and arr[i] = 2 are $\frac{1}{3}$ since all are equally likely to occur.So finding average we should consider three cases and add them up and multiply with probability(1/3)

1. $arr[i] = 0$

$W_0(n)$ is the case when elements in the array is 0

$$W_0(n) = \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} \sum_{k=0}^{\lfloor \log_3 n \rfloor + 2} 1$$

$$\sum_{k=0}^{\lfloor \log_3 n \rfloor} 1 = \lfloor \log_3 n \rfloor + 2$$

$$\sum_{j=i}^{n-1} \lfloor \log_3 n \rfloor + 2 = (\lfloor \log_3 n \rfloor + 2) \cdot (n - i)$$

$$\sum_{i=0}^{n-1} ((\lfloor \log_3 n \rfloor + 2) \cdot (n - i))$$

$$= \sum_{i=0}^{n-1} ((\lfloor \log_3 n \rfloor + 2) \cdot n) - \sum_{i=0}^{n-1} (i \cdot (\lfloor \log_3 n \rfloor + 2))$$

$$= (\lfloor \log_3 n \rfloor + 2) \cdot n \cdot n - (\lfloor \log_3 n \rfloor + 2) \cdot \frac{n \cdot (n-1)}{2}$$

$$= \frac{(\lfloor \log_3 n \rfloor + 2)n^2}{2} + \frac{n(\lfloor \log_3 n \rfloor + 2)}{2}$$

2. $arr[i] = 2$

$W_2(n)$ is the case when elements in the array is 2

$$W_2(n) = \sum_{i=0}^{n-1} \sum_{p=0}^{n-1} 1$$

$$\sum_{p=0}^{p=n-1} 1 = n$$

$$= \sum_{i=0}^{n-1} n$$

$$= n^2$$

$$
\begin{aligned}
A\,(n) &= (1/3)\cdot(W_0(n)+W_2(n)+\sum_{i=0}^{n-1}0)\\
&= (1/3)\cdot[(\frac{(\lfloor\log_3 n\rfloor+2)n^2}{2}+\frac{n(\lfloor\log_3 n\rfloor+2)}{2}+n^2]\\
&= (1/6)\cdot[(\lfloor\log_3 n\rfloor+2)n^2+2n^2+n(\lfloor\log_3 n\rfloor+2)]\\
A\,(n) &= \Theta(n^2\cdot\log n)
\end{aligned}
$$

1.4 Basic operations are the three assignments marked as (4)

1.4.1 Analysis of B(n)

$\rightarrow$ To find best case,we should take the min of cases which is according to the values of elements in array are 0,1,2 respectively.Therefore, best case occurs when the array consists of only 0's

$$
\begin{aligned}
B\,(n) &= \min\{W_0(n),W_1(n),W_2(n)\}\\
&= W_0(n)\\
&= \frac{(\lfloor\log_3 n\rfloor+1)n^2}{2}+\frac{n(\lfloor\log_3 n\rfloor)+1}{2}\\
B\,(n) &= \Theta(n^2\cdot\log n)
\end{aligned}
$$

1.4.2 Analysis of W(n)

$\rightarrow$ To find worst case elements of array can be 1 or 2 but there is different number of execution for different values of i.Let's compare the number of execution when $arr[i]$ is 1 or 2 to find the maximum of number of execution.We should find the i value that makes both side equal.After that, in worst case, array elements should be 1 until the array[i].

$$
\begin{aligned}
W(n) &= \sum_{i=0}^{n-1}(\max\left\{\sum_{j=i}^{n-1}\sum_{k=0}^{\lfloor\log_3 n\rfloor+1}1,\sum_{m=i}^{n-1}\sum_{l=m}^{n-1}\sum_{t=1}^{n}\lfloor\frac{n}{t}\rfloor,\sum_{p=0}^{n-1}\sum_{j=0}^{p^2-1}1\right\})\\
&= \sum_{i=0}^{n-1}(\max\left\{(\lfloor\log_3 n\rfloor+1)\cdot(n-i),\frac{(n-i)(n-i+1)n\ln(n)}{2},\frac{(n-1)n(2n-1)}{6}\right\})
\end{aligned}
$$

$$\frac{(n-i_e)(n-i_e+1)n\ln(n)}{2} = \frac{(n-1)n(2n-1)}{6}$$

$$i_e^2 - (2n+1)i_e = \frac{2n^2-3n+1}{3\ln(n)} - n^2 - n$$

$$(i_e - (n+\frac{1}{2}))^2 = \frac{2n^2-3n+1}{3\ln(n)} + \frac{1}{4}$$

$$i_e = n - \left\lfloor \sqrt{\frac{2n^2-3n+1}{3\ln(n)} + \frac{1}{4}} - \frac{1}{2} \right\rfloor$$

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

$$W(n) = \left[\frac{(i_e+1)(n^2+n)}{2} - \frac{i_e(i_e+1)(2n+1)}{4} + \frac{(i_e)(i_e+1)(2i_e+1)}{12}\right] + \frac{(n-i_e-1)(n^2-n)(2n-1)}{6}$$

$$n - \left\lfloor \sqrt{\frac{2n^2-3n+1}{3\ln(n)} + \frac{1}{4}} - \frac{1}{2} \right\rfloor \leq i_e \leq n$$

$$i_e \sim n; bylimittest$$

$$W(n) = \left[\frac{(n+1)(n^2+n)}{2} - \frac{(n^2+n)(2n+1)}{4} + \frac{n(n+1)(2n+1)}{12}\right] n\ln + \frac{(n-n-1)(n^2-n)(2n-1)}{6}$$

$$W(n) = \frac{(2n^3+3n^2+6n+1)n\ln(n)}{12} - \frac{2n^3-3n^2+n}{6}$$

$$W(n) = \Theta(n^4 \log n)$$

1.4.3 Analysis of A(n)

$$A\left(n\right) = \frac{W_0(n) + W_1(n) + W_2(n)}{3}$$

$$W_0(n) = \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} \sum_{k=0}^{\lfloor \log_3 n \rfloor + 1} 1$$

Now, let's break it down step by step:

$$\sum_{k=0}^{\lfloor \log_3 n \rfloor + 1} 1 = \lfloor \log_3 n \rfloor + 1$$

$$\sum_{j=i}^{n-1} \lfloor \log_3 n \rfloor + 1 = (\lfloor \log_3 n \rfloor + 1) \cdot (n - i)$$

$$\sum_{i=0}^{n-1} (\lfloor \log_3 n \rfloor + 1) \cdot (n - i)$$

$$= \sum_{i=0}^{n-1} ((\lfloor \log_3 n \rfloor + 1) \cdot n) - \sum_{i=0}^{n-1} (i \cdot (\lfloor \log_3 n \rfloor + 1))$$

$$= (\lfloor \log_3 n \rfloor + 1) \cdot n \cdot n - (\lfloor \log_3 n \rfloor + 1) \cdot \frac{n \cdot (n-1)}{2}$$

$$= \frac{(\lfloor \log_3 n \rfloor + 1) n^2}{2} + \frac{n(\lfloor \log_3 n \rfloor + 1)}{2}$$

$$W_2(n) = \sum_{i=0}^{n-1} \sum_{p=0}^{n-1} \sum_{j=0}^{p^2-1} 1$$

$$= \sum_{i=0}^{n-1} \sum_{p=0}^{n-1} p^2$$

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

$$= \frac{n^2(n-1)(2n-1)}{6}$$

$$= \frac{2n^4 - 3n^3 + n^2}{6}$$

$$W_1(n) = \sum_{i=0}^{n-1} \sum_{m=i}^{n-1} \sum_{l=m}^{n-1} \sum_{t=1}^{n} \lfloor \frac{n}{t} \rfloor$$

$$\int_1^{n+1} \lfloor \frac{n}{x} \rfloor \, dx \leq \sum_{t=1}^{n} \lfloor \frac{n}{t} \rfloor \leq \int_0^{n} \lfloor \frac{n}{x} \rfloor \, dx$$

$$\int_1^{n+1} \lfloor \frac{n}{x} \rfloor \, dx \leq n + \sum_{t=2}^{n} \lfloor \frac{n}{t} \rfloor \leq n + \int_1^{n} \lfloor \frac{n}{x} \rfloor \, dx$$

$$n \ln\lfloor (n+1) \rfloor \leq \sum_{t=1}^{n} \lfloor \frac{n}{t} \rfloor \leq n + n \ln\lfloor n \rfloor$$

$$\lim_{n\to\infty} \frac{n(1+\ln(n))}{n\ln(n+1)} = \lim_{n\to\infty} \frac{1+\ln(n)}{\ln(n+1)} = \lim_{n\to\infty} \frac{\frac{1}{n}}{\frac{1}{n+1}} = 1$$

$$\sum_{t=1}^{n} \lfloor \frac{n}{t} \rfloor \in\sim n \cdot \ln\lfloor n \rfloor$$

$$\begin{aligned}
W_1(n) &= \sum_{i=0}^{n-1} \sum_{m=i}^{n-1} \sum_{l=m}^{n-1} n \cdot \ln\lfloor n \rfloor \\
&= \sum_{i=0}^{n-1} \sum_{m=i}^{n-1} (n-m) \cdot n \cdot \ln\lfloor n \rfloor \\
&= \sum_{i=0}^{n-1} (n(n-i) - \frac{n(n-1) - i(i-1)}{2}) \cdot n \cdot \ln\lfloor n \rfloor \\
&= \sum_{i=0}^{n-1} (\frac{n^2+n}{2} - i(n+\frac{1}{2}) + \frac{i^2}{2}) \cdot n \cdot \ln\lfloor n \rfloor \\
&= [\frac{n(n^2+n)}{2} - \frac{(2n+1)(n-1)n}{4} + \frac{(n-1)n(2n-1)}{12}] \cdot n \cdot \ln\lfloor n \rfloor \\
&= \frac{(2n^4+6n^3+4n^2)}{12} \cdot \ln\lfloor n \rfloor
\end{aligned}$$

$$\begin{aligned}
A\,(n) &= \frac{W_0(n) + W_1(n) + W_2(n)}{3} \\
&= \frac{\left(\frac{(\lfloor \log_3 n \rfloor + 1) \cdot n^2}{2} + \frac{n \cdot (\lfloor \log_3 n \rfloor + 1)}{2}\right) + \left(\frac{(2n^4+6n^3+4n^2)}{12} \cdot \ln\lfloor n \rfloor\right) + \left(\frac{2n^4 - 3n^3 + n^2}{6}\right)}{3} \\
&= \left(\frac{(2n^4+6n^3+10n^2+6n)}{36} \cdot \ln\lfloor n \rfloor\right) + \left(\frac{2n^4 - 3n^3 + 4n^2 + 3n}{18}\right)
\end{aligned}$$

$$A\,(n) = \theta(n^4 \cdot \log n)$$

2 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) (1): Basic operation is the comparison between the current element in array and 0 (arr[i]==0).
(2): Basic operation is the assignment operation (y<-y+1) for both marked lines.
(3): Basic operation is the comparison between k and 0 (k<0) for the first marked line and p and n (p<n) for the second marked line.
(4): Basic operation is the assignment operation (y<-y+1) for both marked lines.

3 Real Execution

N Size	Time Elapsed
1	0.00000
5	0.00001
10	0.00004
20	0.00014
30	0.00040
40	0.00068
50	0.00136
60	0.00059
70	0.00102
80	0.00104
90	0.00167
100	0.00205
110	0.00234
120	0.00567
130	0.00564
140	0.00372
150	0.00756

Table 1: Best Case Real Execution Times

N Size	Time Elapsed
1	0.00000
5	0.00000
10	0.01569
20	0.01556
30	0.07929
40	0.28182
50	0.62802
60	1.27174
70	2.16729
80	3.91055
90	5.89472
100	9.28840
110	13.9979
120	19.8523
130	27.4566
140	36.5417
150	47.2357

Table 2: Worst Case Real Execution Times

N Size	Time Elapsed
1	0.00000
5	0.00000
10	0.00000
20	0.00891
30	0.03923
40	0.09433
50	0.26387
60	0.48136
70	0.91832
80	1.55705
90	2.29299
100	3.73899
110	5.06860
120	6.88992
130	10.8757
140	14.8489
150	19.0528

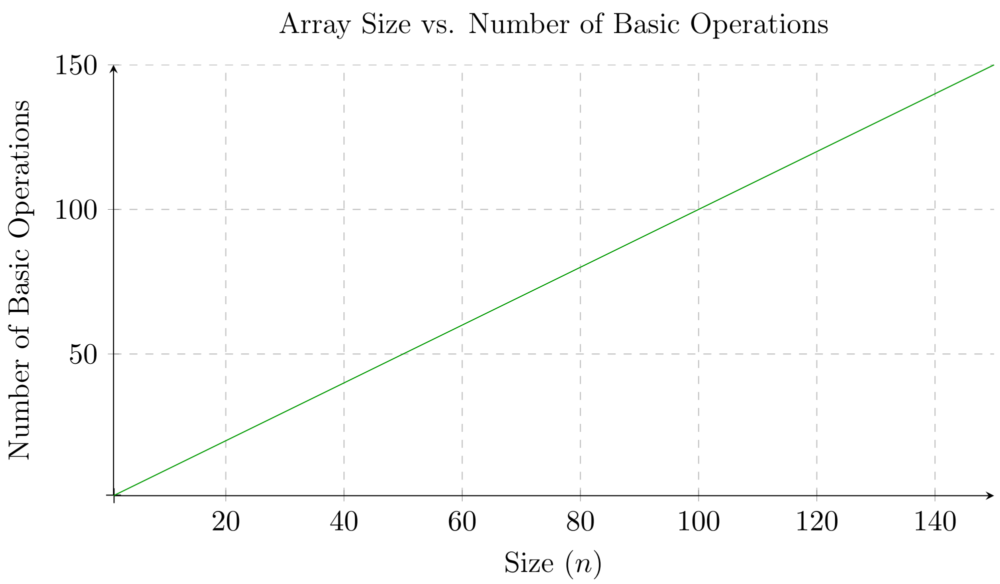
Table 3: Average Case Real Execution Times

4 Comparison

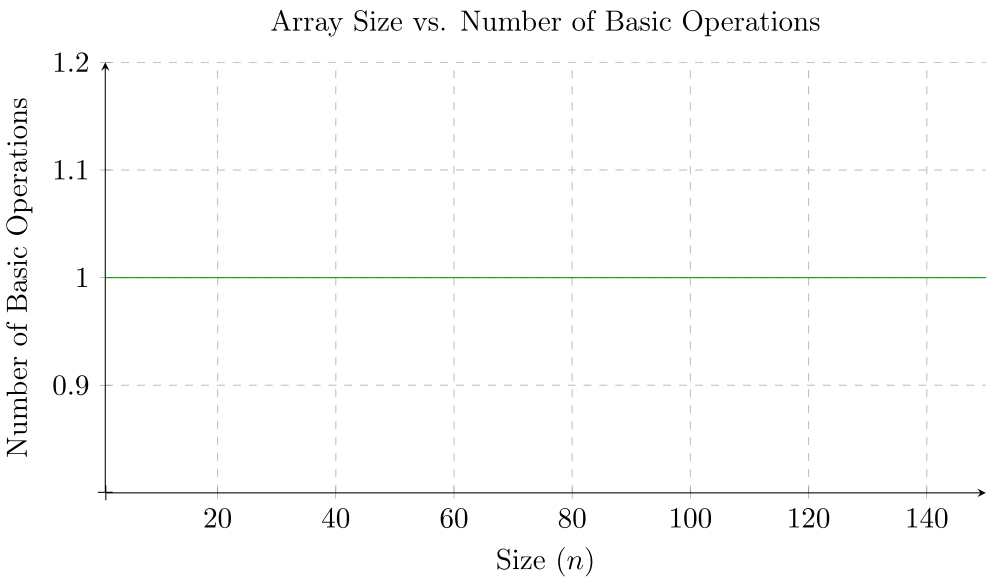
4.1 Best Case

4.1.1 Graph of the real execution time of the algorithm

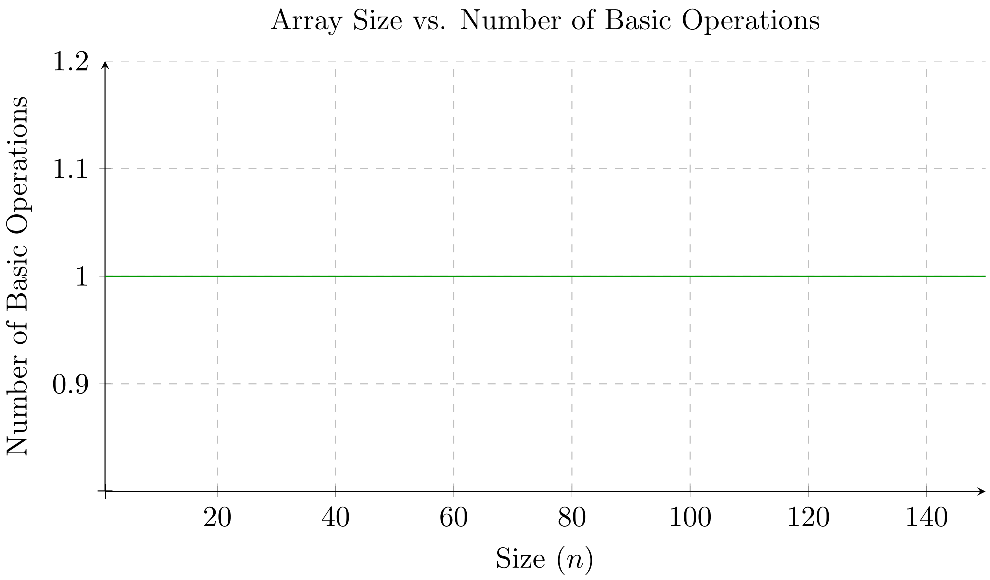
4.1.2 Graph of the theoretical analysis when basic operation is the operation marked as (1)



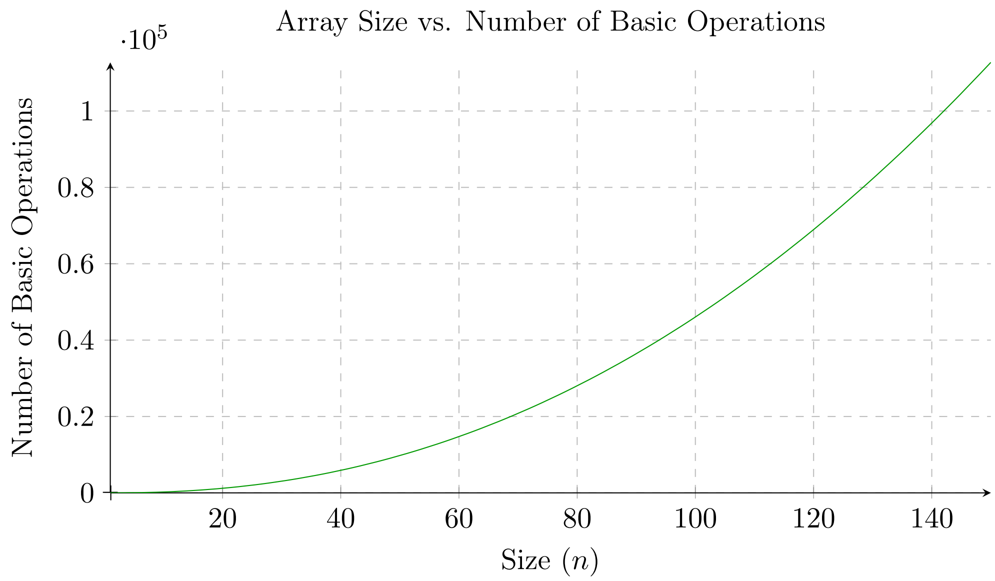
4.1.3 Graph of the theoretical analysis when basic operation is the operation marked as (2)



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



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

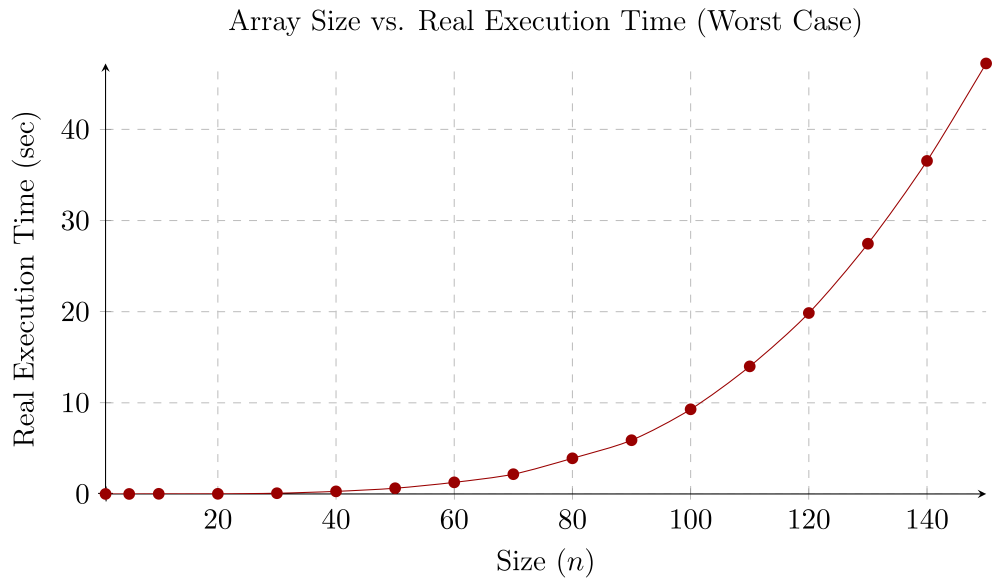


4.1.6 Comments

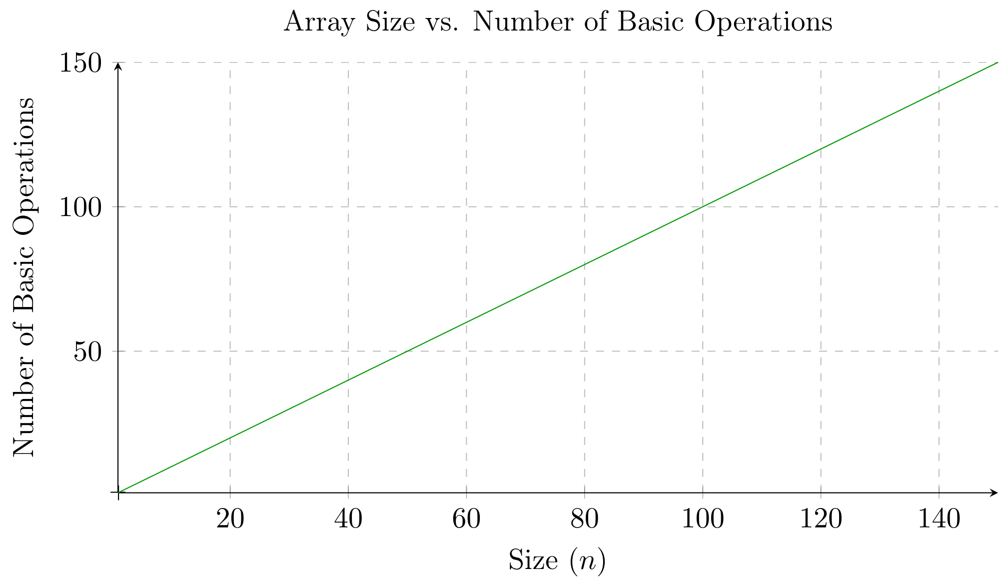
Firstly, we can say that the graph of the real execution time of the algorithm for the best case has an increasing growth rate which means the number of basic operations executed increases exponentially even though execution times fluctuate at some points. The basic operation marked as (1) has a linear growth rate, so it is understood that operation marked (1) is not the basic operation of the algorithm. In the same way, we can understand operations marked (2) and (3) are not the basic operation since they have constant growth rate in their graphs. Lastly, basic operation (4) has an exponential growth rate in its graph of the theoretical analysis. Thus, we can say that operation marked as (4) is the basic operation of the algorithm.

4.2 Worst Case

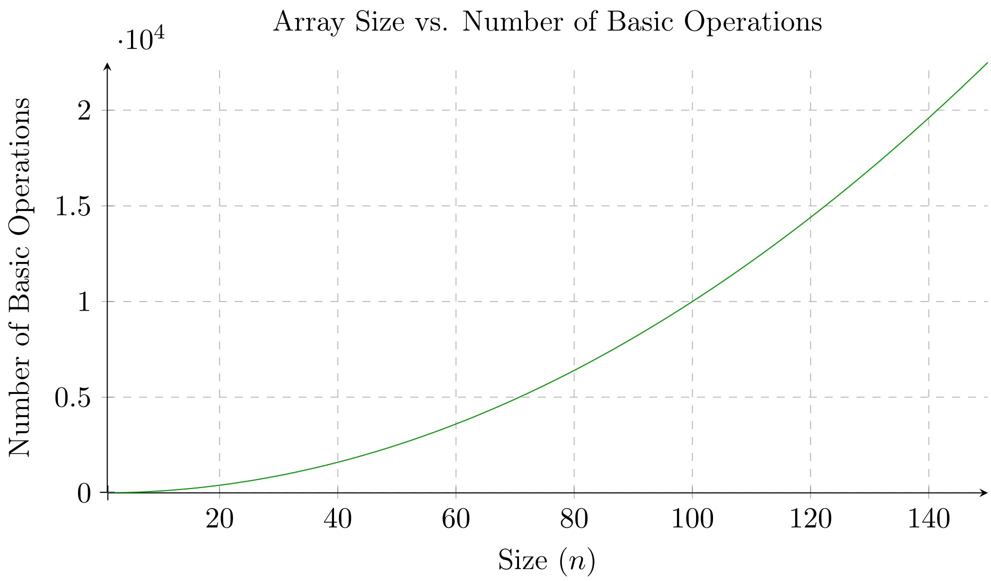
4.2.1 Graph of the real execution time of the algorithm



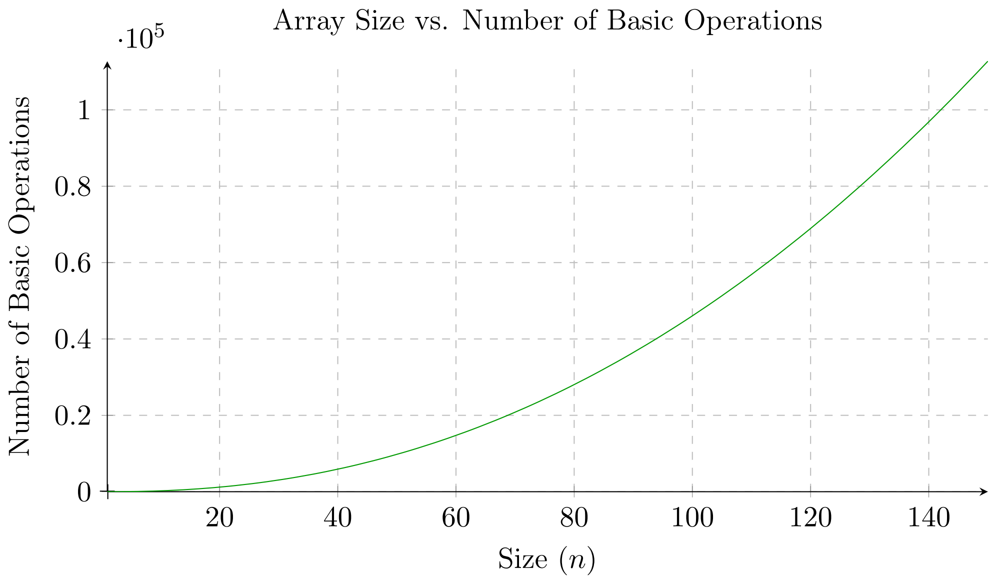
4.2.2 Graph of the theoretical analysis when basic operation is the operation marked as (1)



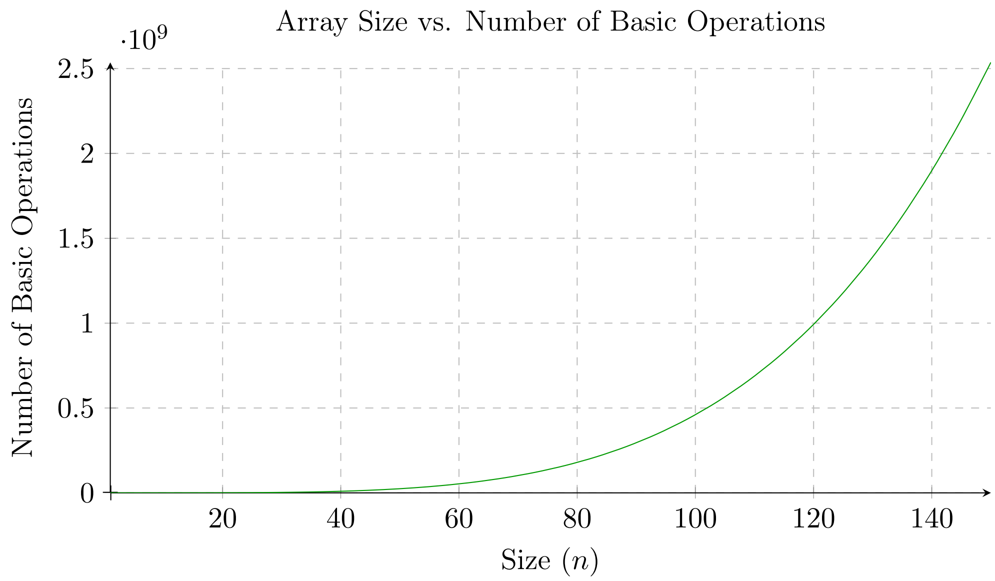
4.2.3 Graph of the theoretical analysis when basic operation is the operation marked as (2)



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



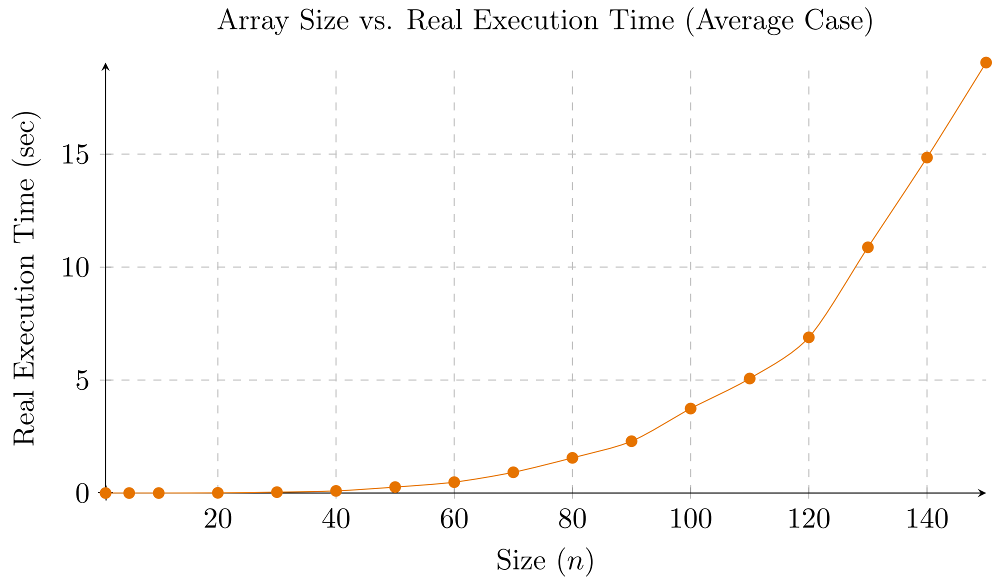
4.2.5 Graph of the theoretical analysis when basic operation is the operation marked as (4)



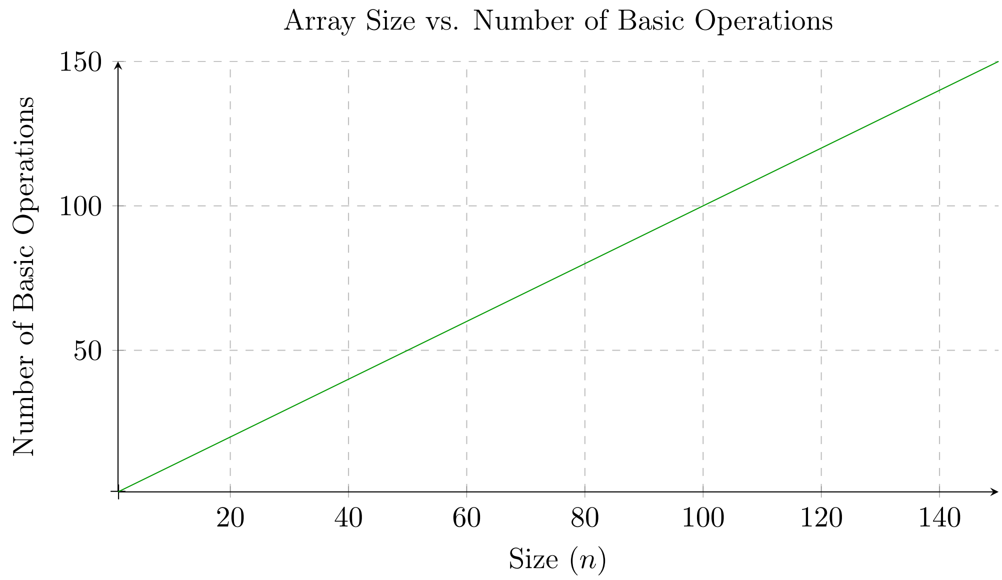
4.2.6 Comments

Firstly, we can say that the graph of the real execution time of the algorithm for the worst case has a strongly increasing growth rate with no fluctuations which means the number of basic operations executed increases exponentially. The basic operation marked as (1) has linear growth rate so we can clearly see it is not the basic operation of the algorithm. The operations marked as (2) and (3) have exponential growth rate so "Array Size vs. Real Execution Time (Worst Case)" graph resembles "Array Size vs. Number of Basic Operations" graph of these two operations. In addition, basic operation (4) also has exponentially growth rate in its graph and we can say the basic operation marked as (4) is much similar with real execution graph. To sum up, we can not easily determine the basic operation looking only graphs. We need to analyze the algorithm theoretically.

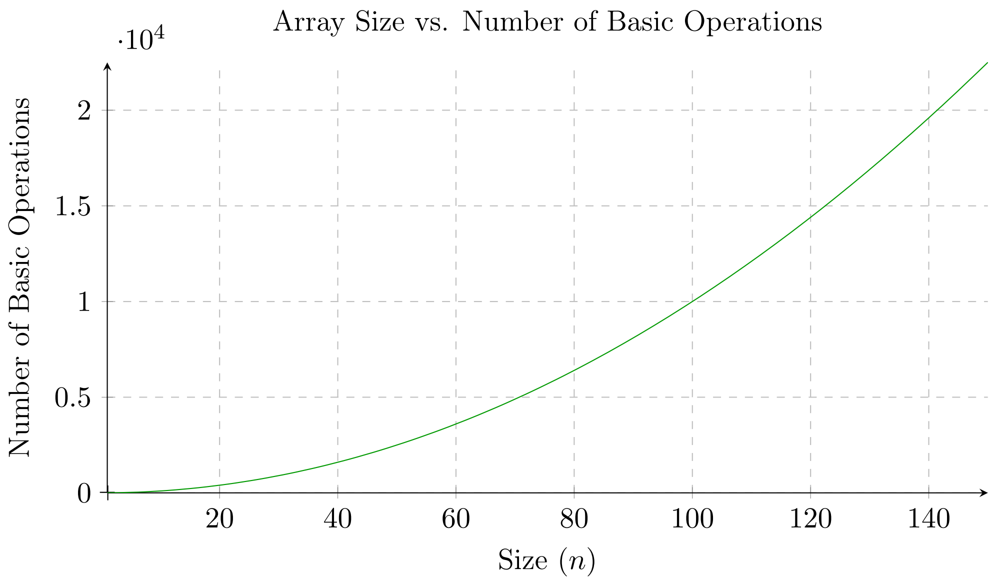
4.3 Average Case



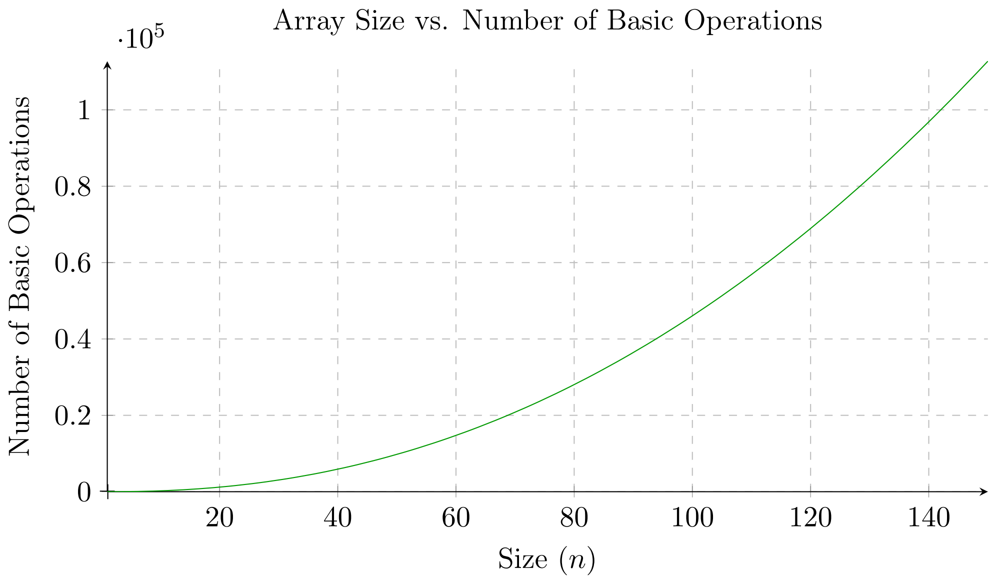
4.3.1 Graph of the theoretical analysis when basic operation is the operation marked as (1)



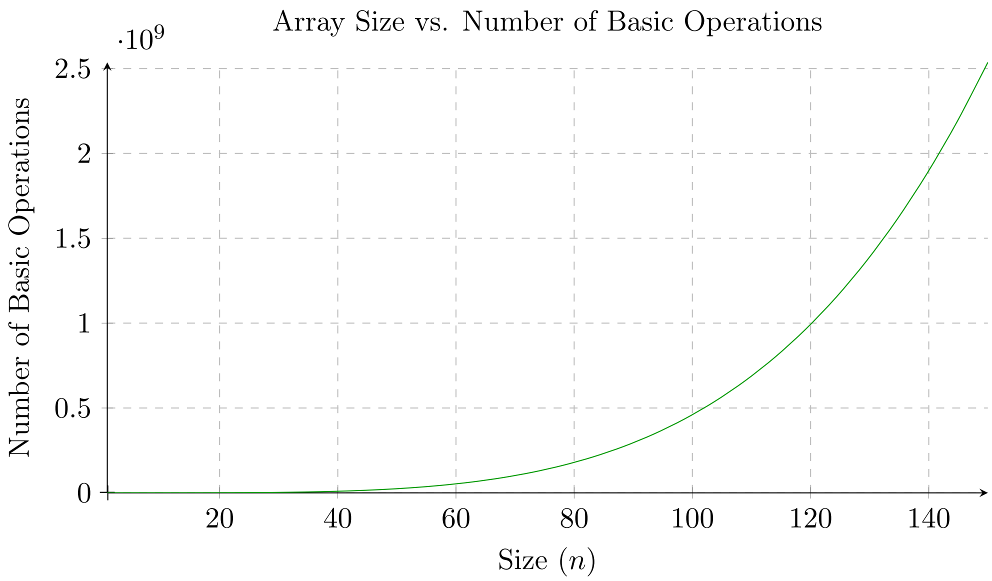
4.3.2 Graph of the theoretical analysis when basic operation is the operation marked as (2)



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



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



4.3.5 Comments

Firstly, we can say that the graph of the real execution time of the algorithm for the average case has a strongly increasing growth rate with some small fluctuations, although we executed every input size with 10 times, which means the number of basic operations executed increases exponentially. The basic operation marked as (1) has linear growth rate so we can clearly see it is not the basic operation of the algorithm. The operations marked as (2) and (3) have exponential growth rate so "Array Size vs. Real Execution Time (Average Case)" graph resembles "Array Size vs. Number of Basic Operations" graph of these two operations. In addition, basic operation (4) also has exponentially growth rate in its graph and we can say the basic operation marked as (4) is much similar with real execution graph but we can not confirm this categorically. To sum up, we can not easily determine the basic operation looking only graphs. We need to analyze the algorithm theoretically.