Algorithms: Practical 1 Algorithm Analysis

18400034 Noor Bari

1. Algorithms for Multiplication

Multiply 68 x 139

Half	Double
~~68~~	~~139~~
~~34~~	~~278~~
17	556
~~8~~	~~1112~~
~~4~~	~~2224~~
~~2~~	~~4448~~
1	8896

556 + 8896 = 9452

2. Counting Instructions

- Assigning a value to a variable
- Calling a method
- Performing an arithmetic operation
- Comparing two numbers
- Indexing into an array
- Following an object reference
- Returning from a method

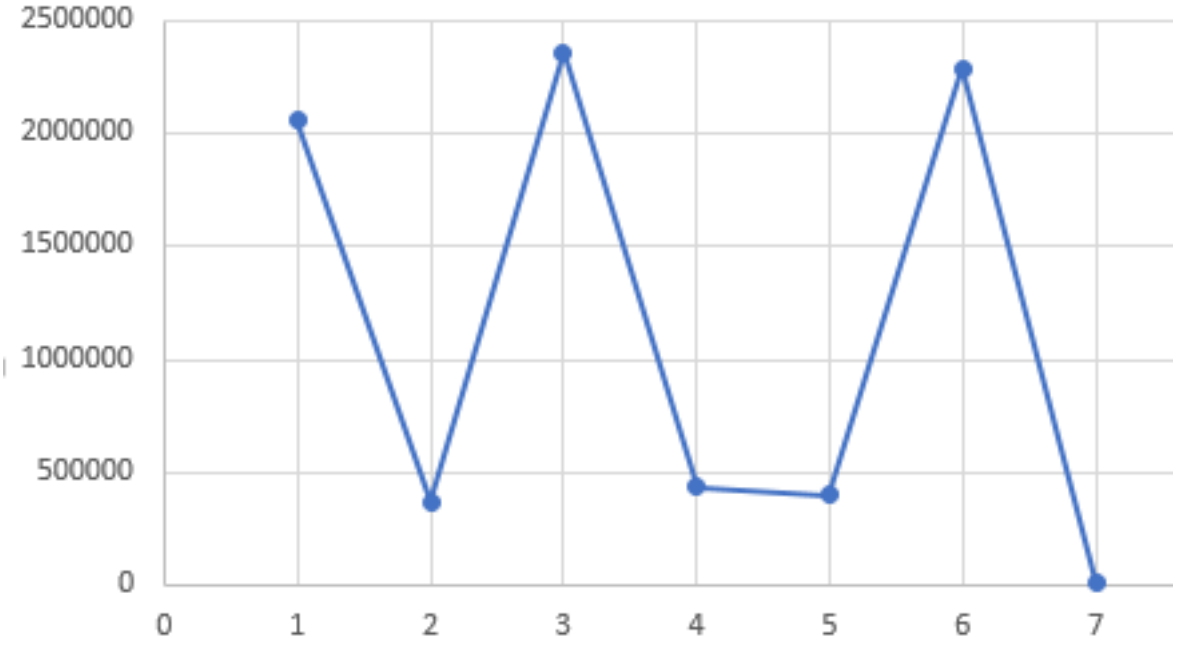
arrayMax function	Finds the biggest integer in an array
Input: an array A of N integers	
Output: maximum element of A	
arrayMax(A, n) {	
currentMax = A[0]	2
For(i=0; i<A.length; i++){	4N - 2
If(A[i] > currentMax) then	2(N-1)
currentMax = A[i]	2(N-1)
}	
Return currentMax	1
End function	Total: T(N) = 8N - 2

3. Implementing the Russian Peasant's algorithm in Java (using ints/ longs) and verify its correctness.

```
public class PracticalOne {

	public static int russianMultiply(int a, int b) {
		int res = 0;

		while (b > 0) {

			if (b % 2 != 0) {
				res += a;
			}

			a = a * 2;
			b = b / 2;

		}
		return res;
	}

	public static void main(String[] args) {
		int a = 60000568;
		int b = 1390000000;

		final long startTime = System.nanoTime();
		System.out.println(russianMultiply(a, b));
		final long elapsedTime = System.nanoTime() - startTime;
		System.out.println("Time taken: " + elapsedTime);

	}
}
```

Input	Time
(2, 3)	2061170
(22, 33)	369672
(222, 333)	2364295
(2222, 3333)	432939
(22222, 33333)	394979
(222222, 333333)	2289752
(2222222, 3333333)	8896

Graph:


2500000
2000000
1500000
1000000
500000
0
0
1
2
3
4
5
6
7