EECS 233 HW8

Ben Pierce
bgp12@case.edu

November 5, 2017

GitHub: https://github.com/bp0017/CWRUEECS233/tree/master/HW8

1 Question 1: Program Output

1.1 A

```
C:\Users\bp001\Documents\EECS223\HW8>java BinarySearcher
Searching for numbers in an array.
Is -1 in the array? No.
Is 0 in the array? No.
Is 1 in the array? No.
Is 2 in the array? Yes, at index [0].
Is 3 in the array? No.
Is 4 in the array? Yes, at index [1].
Is 5 in the array? No.
Is 6 in the array? Yes, at index [2].
Is 7 in the array? No.
Is 8 in the array? Yes, at index [3].
Is 9 in the array? No.
Is 10 in the array? Yes, at index [4].
Is 11 in the array? No.
Is 12 in the array? Yes, at index [5].
Is 13 in the array? No.
Is 14 in the array? Yes, at index [6].
Is 15 in the array? No.
Is 16 in the array? No.
Searching for 0 in an empty array:  Not found.
End of searching.
```

1.2 B

```
Searching for numbers in an array.
Is -1 in the array? Searching before 3 Searching before 1
Searching before 0 No.
Is 0 in the array? Searching before 3 Searching before 1
Searching before 0 No.
Is 1 in the array? Searching before 3 Searching before 1
Searching before 0 No.
Is 2 in the array? Searching before 3 Searching before 1
Found at index 0
Is 3 in the array? Searching before 3 Searching before 1
Searching after 0 No.
Is 4 in the array? Searching before 3 Found at index 1
Is 5 in the array? Searching before 3 Searching after 1
Searching before 2 No.
Is 6 in the array? Searching before 3 Searching after 1
Found at index 2
Is 7 in the array? Searching before 3 Searching after 1
Searching after 2 No.
Is 8 in the array? Found at index 3
Is 9 in the array? Searching after 3 Searching before 5
Searching before 4 No.
Is 10 in the array? Searching after 3 Searching before 5
Found at index 4
Is 11 in the array? Searching after 3 Searching before 5
Searching after 4 No.
Is 12 in the array? Searching after 3 Found at index 5
Is 13 in the array? Searching after 3 Searching after 5 No.
Is 14 in the array? Searching after 3 Searching after 5 No.
Is 15 in the array? Searching after 3 Searching after 5 No.
Is 16 in the array? Searching after 3 Searching after 5 No.
Searching for 0 in an empty array: Not found.
End of searching.
```

2 Question 2

2.1 A

```
[1,3,2,4,6,5] Target = 1.
```

2.2 B

If the array was checked to be sorted before the search, the total runtime would be $N + \log_2 N = O(N)$ because the time to verify that an array is sorted is linear time, plus the time to search. This equals $O(N)$ because N time is larger then $\log_2 N$.

3 Question 3

3.1 A

Key	Calculation	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
1	1%13 = 1		1											
2	2%13 = 2		1	2										
12	12%13 = 12		1	2										12
13	13%13=0	13	1	2										12
14	14%13=1	13	1	2	14									12
130	130%13=0	13	1	2	14	130								12
1212	1212%13=3	13	1	2	14	130	1212							12
1301	1301%13=1	13	1	2	14	130	1212	1301						12
1300	1300%13=0	13	1	2	14	130	1212	1301	1300					12

3.2 B

Key	Calculation	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
1	1%13 = 1		1											
2	2%13 = 2		1	2										
12	12%13 = 12													12
13	13%13=0	13	1	2										12
14	14%13=1, 14%11=3	13	1	2		14								12
130	130%13=0, 130%11=9	13	1	2		14					130			12
1212	1212%13=3, 1212%11=2	13	1	2	1212	14					130			12
1301	1301%13=1,1301%11=3	13	1	2	1212	14			1301		130			12
1300	1300%13=0, 1300%11=2	13	1	2	1212	14		1300	1301		130			12

4 Question 4

```
C:\Users\bp001\Documents\EECS223\HW8>java Table2
Trying index [1]
Trying index [2]
Trying index [12]
Trying index [0]
Trying index [1]
Trying index [2]
```

```
Trying index [3]
Trying index [0]
Trying index [1]
Trying index [2]
Trying index [3]
Trying index [4]
Trying index [12]
Trying index [0]
Trying index [1]
Trying index [2]
Trying index [3]
Trying index [4]
Trying index [5]
Trying index [1]
Trying index [2]
Trying index [3]
Trying index [4]
Trying index [5]
Trying index [6]
Trying index [0]
Trying index [1]
Trying index [2]
Trying index [3]
Trying index [4]
Trying index [5]
Trying index [6]
Trying index [7]
[0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]
13 1 2 14 130 1312 1301 1300 null null null null 12
```

5 Question 5

```
C:\Users\bp001\Documents\EECS223\HW8>java DoubleHash
Trying index [1] for 1
Trying index [2] for 2
Trying index [12] for 12
Trying index [0] for 13
Trying index [1] for 14
Trying index [4] for 14
Trying index [0] for 130
Trying index [9] for 130
```

```
Trying index [3] for 1212
Trying index [1] for 1301
Trying index [4] for 1301
Trying index [7] for 1301
Trying index [0] for 1300
Trying index [2] for 1300
Trying index [4] for 1300
Trying index [6] for 1300
[0][1][2][3][4][5][6][7][8][9][10][11][12]
13 1 2 1212 14 null 1300 1301 null 130 null null 12
```

6 Question 6

6.1 A

The run time for adding N values to a hash table with no collisions is $O(N)$, because each value in N is being inserted with a single assignment operation, so the total runtime is $1 * N$ or $O(N)$

6.2 B

The runtime of finding a single value to a hash table with no collisions is $O(1)$, because it is a single operation. This is because the hash value is the index where the key is found, due to the assumption of no collisions.