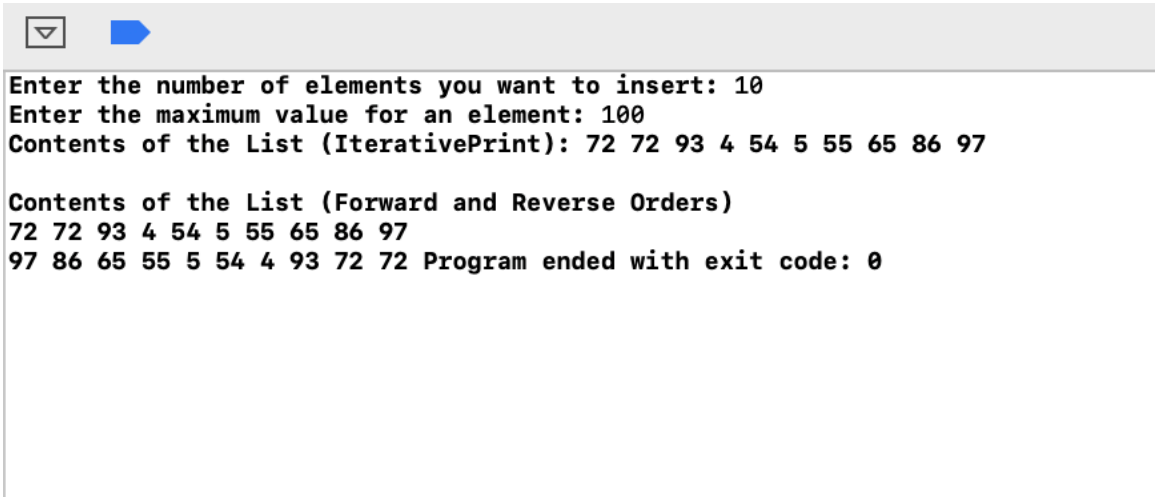
Kyla Wilson
Feb 20, 2019
J00813814

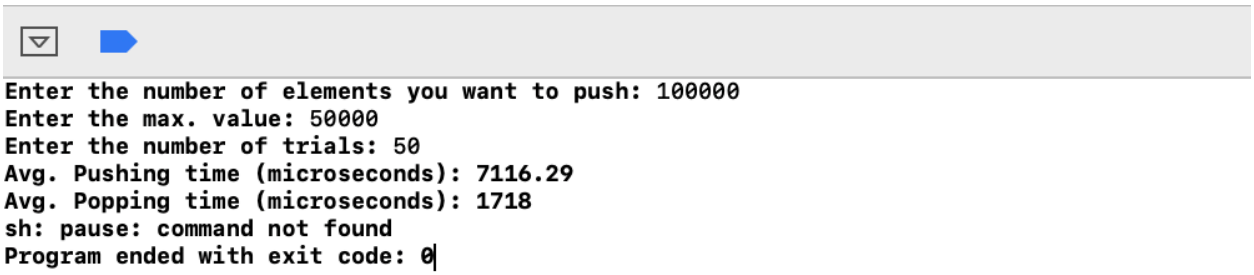
Q1



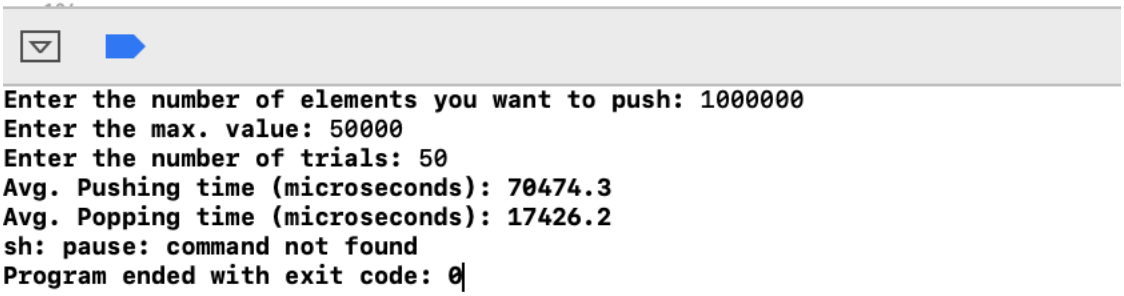
```
Enter the number of elements you want to insert: 10
Enter the maximum value for an element: 100
Contents of the List (IterativePrint): 72 72 93 4 54 5 55 65 86 97

Contents of the List (Forward and Reverse Orders)
72 72 93 4 54 5 55 65 86 97
97 86 65 55 5 54 4 93 72 72 Program ended with exit code: 0
```

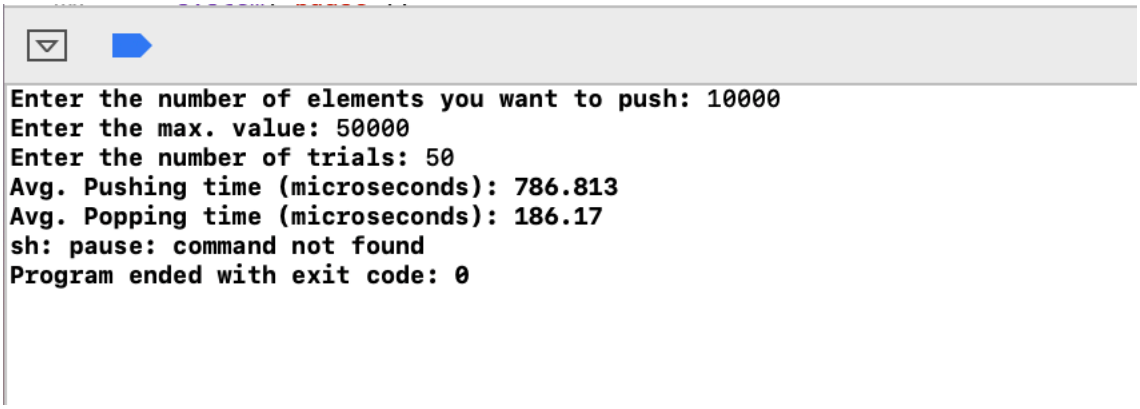
Q2



```
Enter the number of elements you want to push: 100000
Enter the max. value: 50000
Enter the number of trials: 50
Avg. Pushing time (microseconds): 7116.29
Avg. Popping time (microseconds): 1718
sh: pause: command not found
Program ended with exit code: 0
```

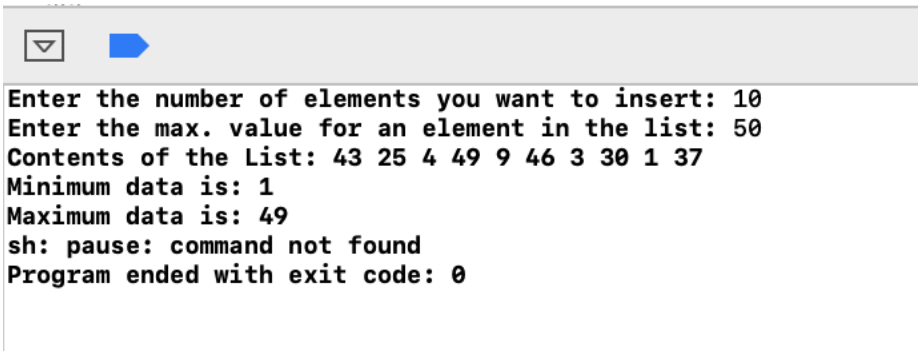


```
Enter the number of elements you want to push: 1000000
Enter the max. value: 50000
Enter the number of trials: 50
Avg. Pushing time (microseconds): 70474.3
Avg. Popping time (microseconds): 17426.2
sh: pause: command not found
Program ended with exit code: 0
```



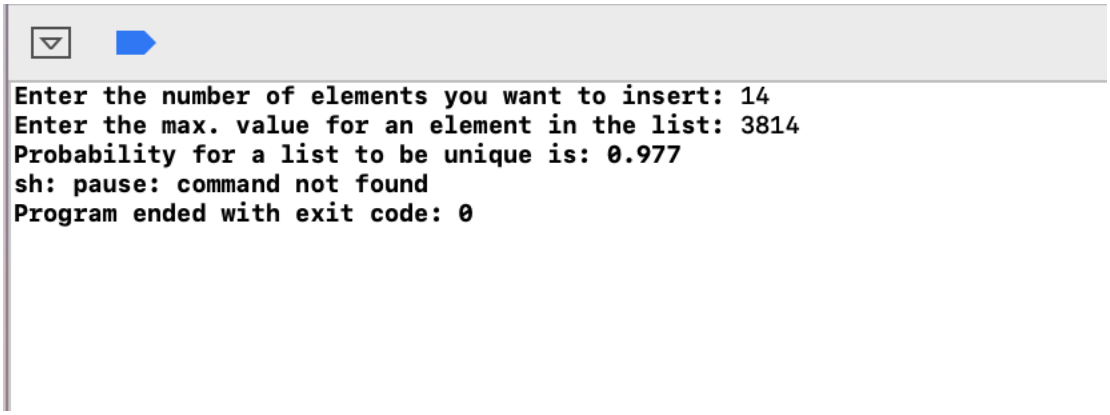
```
Enter the number of elements you want to push: 10000
Enter the max. value: 50000
Enter the number of trials: 50
Avg. Pushing time (microseconds): 786.813
Avg. Popping time (microseconds): 186.17
sh: pause: command not found
Program ended with exit code: 0
```

Q3



```
Enter the number of elements you want to insert: 10
Enter the max. value for an element in the list: 50
Contents of the List: 43 25 4 49 9 46 3 30 1 37
Minimum data is: 1
Maximum data is: 49
sh: pause: command not found
Program ended with exit code: 0
```

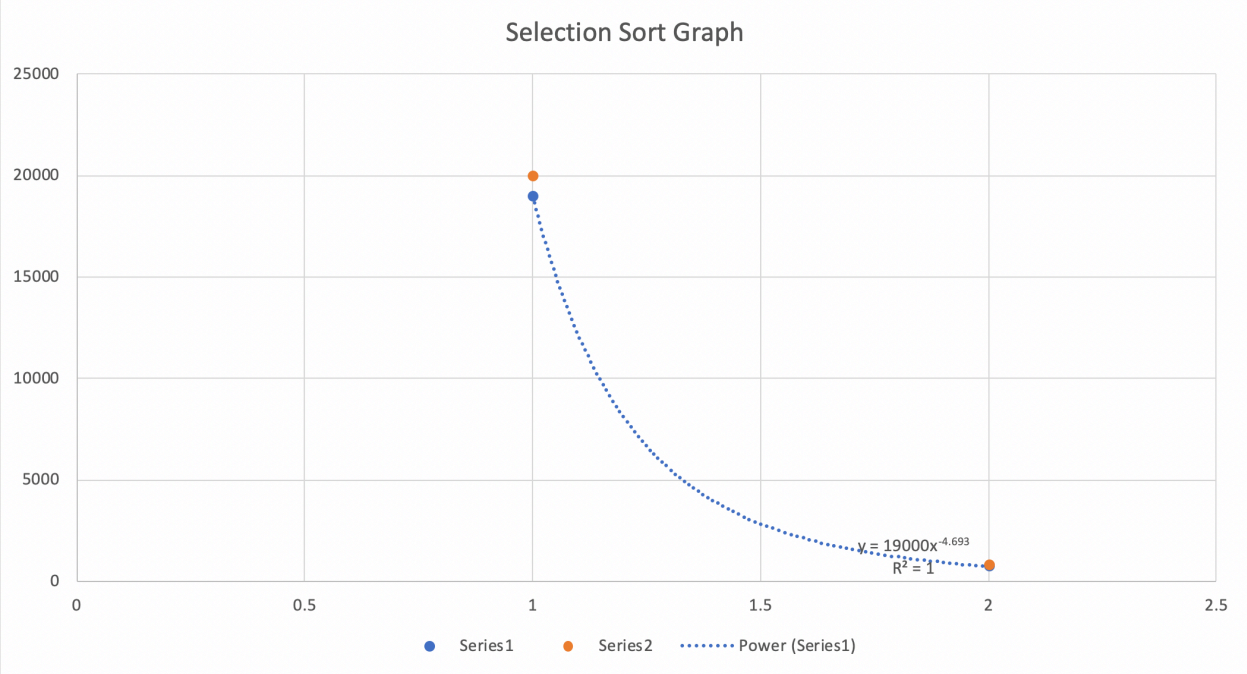
Q4



```
Enter the number of elements you want to insert: 14
Enter the max. value for an element in the list: 3814
Probability for a list to be unique is: 0.977
sh: pause: command not found
Program ended with exit code: 0
```

X	Y
1000	2.1409
2000	8.19345
3000	18.2309
4000	31.8665
5000	50.3696
6000	72.311
7000	98.8214
8000	128.741
9000	162.139
10000	201.477
11000	245.088
12000	293.126
13000	339.793
14000	408.31
15000	461.569
16000	515.009
17000	575.763
18000	649.562
19000	734.675
20000	810.791

Q5



Algorithm Comparison:

The time complexity of my selection sort algorithm is O(n^2) due to the double for-loop. The R^2 equation equals one, therefore I assume it takes approximately one minute to run the code. The Y equal value seems random to me and I don't see how it has something to do with the time complexity of the code.c