Michael Mroczka

Bruce Elenbogen - CIS 3501

Program 4: Backtracking — In Your Prime (Find the Prime Sequence)

**Big-O SPEED Analysis per Function:**

|  |  |  |
| --- | --- | --- |
| **Function** | **Big-O** | **Reasoning/Explanation** |
| [Constructor] Primes(int first, int last) | O(n2) | Because it calls getPrimes |
| void getPrimes() | O(n2) | Due to the nested for-loop. |
| bool isPrime(int n) | O(n) | Due to the single for-loop |
| void recursion(vector<int>, vector<int>) | O(n!) | Since you are looking at the worst case, the worst case is that it recurses for every permutation so technically the worst case would be n! even if the worst case will never truly happen (due to the backtracking and intelligently recognizing when a sequence won’t work |
| void printVec(vector<int> vectorToPrint) | O(n) | Due to the single for-loop |
| main() | O(n!) | Due to it calling the recursion function. |

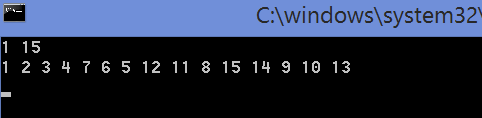
**Big-O SPACE Analysis per Function:**

|  |  |  |
| --- | --- | --- |
| **Function** | **Big-O** | **Reasoning/Explanation** |
| Primes(int first, int last) | O(n) | Due to it calling the getPrimes function. |
| void getPrimes() | O(n) | Due to the push\_back being used in the for-loop |
| bool isPrime(int n) | O(1) | N/A |
| void recursion(vector<int>, vector<int>) | O(n!) | You potentially can make as many as n! new items. This isn’t the true worst case, but rather a theoretical worst case. |
| void printVec(vector<int> vectorToPrint) | O(1) | N/A |
| main() | O(n!) | Potentially as big as n! since you are calling the recursion function |

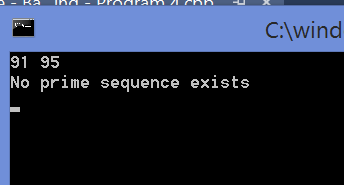
**White Box Test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Being Tested** | **Test** | **Reasoning** | **Input** | **Actual Output** |
| Int main() | Testing basic functionality | Seeing if it can handle a basic test | 1 15 | See below |
| void recursion() | Testing to see if it will recognize that a sequence isn’t a prime sequence | Sometimes sequences aren’t prime and it needs to recognize that and terminate to avoid an infinite loop | 91 95 | See below |
| Int main() | Multiple test cases | To ensure the program resets everything properly and clears data for new sequences to be tested. | 1 5  9 17  50 75  31 35  0 0 | See below |
| Void recursion() | Testing to see if it can handle large numbered sequences when large > 50 | Programs that can’t do difficult sequences take too long and time out | 1 51  1 100  1 1000 | See below |

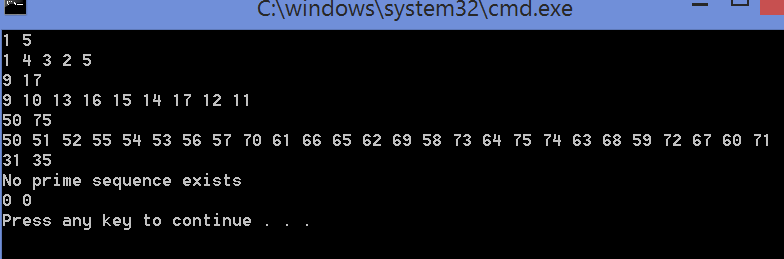
Test 1



Test 2



Test 3



Test 4

