

ottersome / Algorithms

Branch: master

Algorithms / HW2 / ex4.cpp

Find file

Copy path



Resolving conflicts that I had when I worked of my desktop on hw2ex4

60082fe a day ago

1 contributor

121 lines (115 sloc) 4.36 KB

```
1  #include <cstdio>
2  #include <cstring>
3  #define N 23//this means until n = <Your input Number>
4  using namespace std;
5  //so for this algorithm we must try every possible difference in
6  //the array of primes, for each prime and scan O(n/logn) primes
7  //to arrive to our conclusion
8  //we will create the function that will create our prime only
9  //array here
10 //
11 //however here will be our dynamic programming algorithm
12 int A[N+1];
13 const int Asize = sizeof(A)/sizeof(A[0]); //im sorry this is kind of redundant but i build this really fast without much mind to
14 int primeCounter = 0;
15 int currentLargest = 0; //as we create our table this value will change according to the current largest subsequence of primes
16 void buildA(){
17     cout << "Building A : ";
18     for(int i =1;i<=N;i++){
19         A[i-1] = i;
20         cout << A[i-1]<<" ";
21     }
22     cout << endl;
23 }
24 void SieveOfEratosthenes(int n,int primeDictionary[Asize], int primeOnlyArr[Asize])
25 {
26     printf("We will now calculate the primes from 1 to %d\n",n);
27     //this will help us keep track of our
28     //assume all are prime at first, primes are identified if
29     //their index contains 1, not prime = -1
30     memset(primeDictionary, 1,sizeof(A[0])* Asize);
31     primeDictionary[0] = -1;
32     primeDictionary[1] = -1;
33     for (int p=2; p<=n; p++){
34         // If prime[p] is not changed, then it is a prime
35         if (primeDictionary[p] != 1){
36             // Update all multiples of p
37             for (int i=p*2; i<=n; i += p)
38                 primeDictionary[i] = -1;
39         }
40     }
41
42     // Print all prime numbers
43     for (int p=2; p<=n; p++)
44         if (primeDictionary[p])
45             cout << p << " ";
46     cout << endl;
47     //now we index assign the indexes to our array
48     for(int p=2;p<=n;p++){
49         if(primeDictionary[p] != -1){
50             //we found our prime and assign its index
51             //so basically if the element as a number greater
52             //than zero that means its the index to its prime
53             //only array
54             primeDictionary[p] = primeCounter++;
55         }
56     }
57     // Print all prime numbers and store in the prime only array
58     for (int p=2; p<n; p++)
59         if (primeDictionary[p] != -1) {
60             printf("The index for prime %d is %d.\n",p,primeDictionary[p]);
61             primeOnlyArr[primeDictionary[p]] = p;
62         }
```

```

63     cout <<endl;
64 }
65
66 void buildResult(int resultA[][ASize],int primeIndexDictionary[ASize],int primeOnlyArr[ASize]){
67     for(int i = 0;i<primeCounter;i++){
68         for(int j=0;j<ASize; j++){
69             int curPrime = primeOnlyArr[i];//this can be thought of as the rows of result
70             if(j == 0 || curPrime == 2)//these col and row will be populated with 0
71                 continue;
72             //else
73             //fetch previous sequence length
74             if(curPrime - j < 2){
75                 continue;//nothing else to do when we will keep getting differences to < 2
76             }
77             if(primeIndexDictionary[curPrime-j] != -1){
78                 resultA[i][j] += 1 + resultA[primeIndexDictionary[curPrime-j]][j];
79                 if(resultA[i][j] > currentLargest)
80                     currentLargest = resultA[i][j];
81             }
82         }
83     }
84 }
85 void printArray(int result[][ASize],int primeOnlyArr[ASize]){
86     printf("      ");
87     for(int i = 0;i<ASize;i++){
88         printf("[%5d]",i);
89     }
90     cout<<endl;
91     for(int i = 0;i<primeCounter;i++){
92         for(int j = 0;j<ASize;j++){
93             if(j==0){
94                 if(primeOnlyArr[i] == 24)
95                     printf("For some reason primeOnlyArr is %d\n",primeOnlyArr[i]);
96                 printf("[%5d]",primeOnlyArr[i]);
97             }
98             printf("[%5d]",result[i][j]);
99         }
100         cout << endl;
101     }
102 }
103
104 int main(){
105     buildA();
106     int primeIndexDictionary[ASize];//i know it could be less but for simplicity sake
107     int primeOnlyArr[ASize];//same for this one
108
109     SieveOfEratosthenes(ASize,primeIndexDictionary,primeOnlyArr);
110     int result[primeCounter][ASize];
111     memset(result,0,sizeof(result[0][0])*primeCounter*ASize+(1*sizeof(result[0][0])));
112     cout << "Before building the result " <<endl;
113     printArray(result,primeOnlyArr);
114     buildResult(result,primeIndexDictionary,primeOnlyArr);
115     cout <<"After"<<endl;
116     printArray(result,primeOnlyArr);
117     printf("The largest PAP sequence is of %d numbers : \n",currentLargest+1);
118
119     return 0;
120 }

```