

Algorithm Lab 2

Supta Richard Philip

January 31, 2023

1 Pre-Requisite in C++

1. Function
2. Pointer
3. Array / Dynamic Array
4. Dynamic Array - 2D/ Matrix
5. Random number
6. File Operation

2 Warm up problem

1. Prime number
2. Maximum Pairwise Product
3. Find second largest number in a List
4. Linear Search
5. Bubble sort
6. Selection sort
7. Insertion Sort

Algorithm 1 IsPrime

```
1: procedure ISPRIME( $n$ )
2:    $Flag \leftarrow true$ 
3:   for  $i \leftarrow 2, \sqrt{n}$  do
4:     if  $n \% i == 0$  then
5:        $Flag \leftarrow false$ 
6:       break
7:     end if
8:   end for
9:   return  $Flag$ 
10: end procedure
```

Algorithm 2 Maximum Pairwise Product

```
1: procedure MAXPAIRWISEPRODUCTNAIVE( $A, n$ )
2:    $product \leftarrow 0$ 
3:   for  $i \leftarrow 1, n$  do
4:     for  $j \leftarrow 1, n$  do
5:       if  $i \neq j$  then
6:         if  $product < A[i] * A[j]$  then
7:            $product = A[i] * A[j]$ 
8:         end if
9:       end if
10:    end for
11:  end for
12:  return  $product$ 
13: end procedure
```

Algorithm 3 Maximum Pairwise Product Fast

```
1: procedure MAXPAIRWISEPRODUCTFAST( $A, n$ )
2:    $index_1 \leftarrow 0$ 
3:   for  $i \leftarrow 0, n - 1$  do
4:     if  $A[i] \geq A[index_1]$  then
5:        $index_1 = i$ 
6:     end if
7:   end for
8:    $index_2 \leftarrow 0$ 
9:   for  $i \leftarrow 1, n - 1$  do
10:    if  $i \neq index_1 \ \&\& \ A[i] \geq A[index_2]$  then
11:       $index_2 = i$ 
12:    end if
13:  end for
14:  return  $A[index_1] * A[index_2]$ 
15: end procedure
```

Algorithm 4 Linear Search

```
1: procedure LINEAR( $A, n, item$ )
2:   for  $i \leftarrow 0, n - 1$  do
3:     if  $A[i] == item$  then
4:       return  $i$ 
5:     end if
6:   end for
7:   return  $-1$ 
8: end procedure
```

Algorithm 5 Bubble Sort

```
1: procedure BUBBLESORT( $A, n$ )
2:   for  $k \leftarrow 0, n - 1$  do
3:     for  $i \leftarrow 0, n - 1$  do
4:       if  $A[i] > A[i + 1]$  then
5:          $swap(A[i], A[i + 1])$ 
6:       end if
7:     end for
8:   end for
9: end procedure
```

Algorithm 6 Selection Sort

```
1: procedure SELECTIONSORT( $A, n$ )
2:   for  $i \leftarrow 0, n - 1$  do
3:      $iMin \leftarrow i$ 
4:     for  $j \leftarrow i + 1, n - 1$  do
5:       if  $A[j] < A[iMin]$  then
6:          $iMin = j$ 
7:       end if
8:      $swap(A[iMin], A[i])$ 
9:   end for
10: end for
11: end procedure
```

Algorithm 7 Insertion Sort

```
1: procedure INSERTIONSORT( $A, n$ )
2:   for  $j \leftarrow 1, n - 1$  do
3:      $value \leftarrow A[j]$ 
4:      $i \leftarrow j - 1$ 
5:     while  $i > 0 \& A[i] > value$  do
6:        $swap(A[i], A[i + 1])$ 
7:        $i \leftarrow i - 1$ 
8:     end while
9:      $A[i + 1] = value$ 
10:  end for
11: end procedure
```
