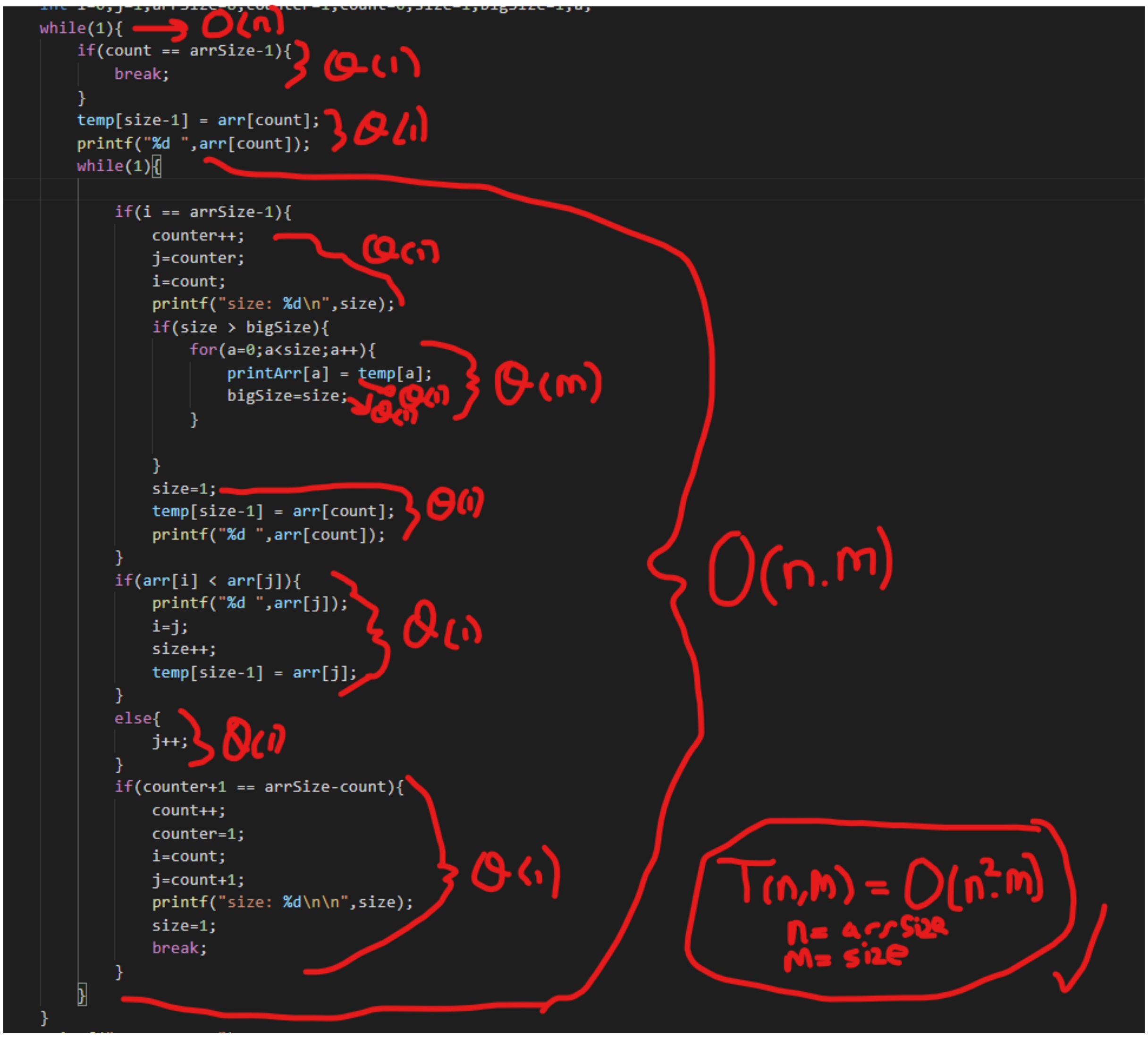
CSE 331 - Spring 2021

Homework 2 Report

Coşkun Hasan ŞALTU
1801042631

1. **Time Complexity of Algorithm**



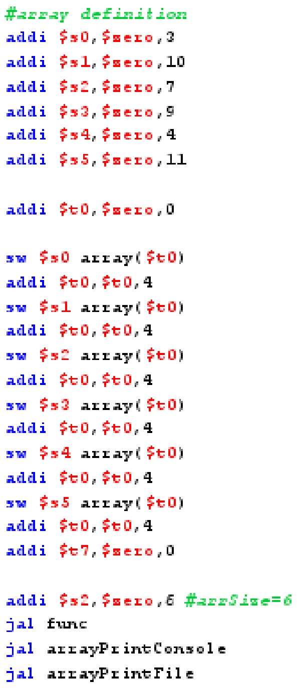
```
while(1){
    if(count == arrSize-1){
        break;
    }
    temp[size-1] = arr[count];
    printf("%d ",arr[count]);
    while(1){

        if(i == arrSize-1){
            counter++;
            j=counter;
            i=count;
            printf("size: %d\n",size);
            if(size > bigSize){
                for(a=0;a<size;a++){
                    printArr[a] = temp[a];
                    bigSize=size;
                }

            }
            size=1;
            temp[size-1] = arr[count];
            printf("%d ",arr[count]);
        }
        if(arr[i] < arr[j]){
            printf("%d ",arr[j]);
            i=j;
            size++;
            temp[size-1] = arr[j];
        }
        else{
            j++;
        }
        if(counter+1 == arrSize-count){
            count++;
            counter=1;
            i=count;
            j=count+1;
            printf("size: %d\n\n",size);
            size=1;
            break;
        }
    }
}
```

2. **Test Case of Algorithm**

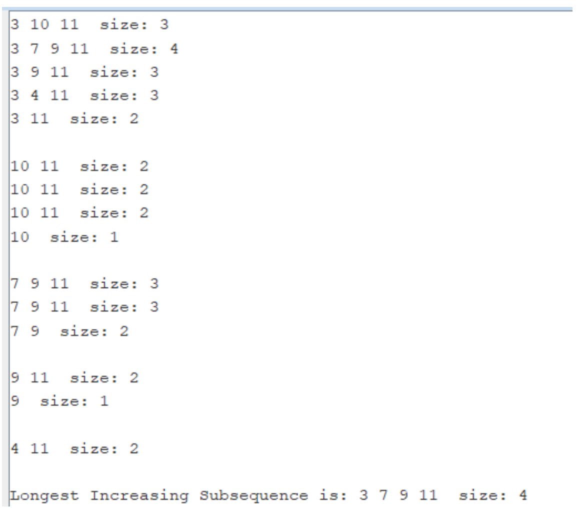
This algorithm is tried with 6 different array.The results printed console and file(testout.txt)

First Array: {3,10,7,9,4,11}

Console Output

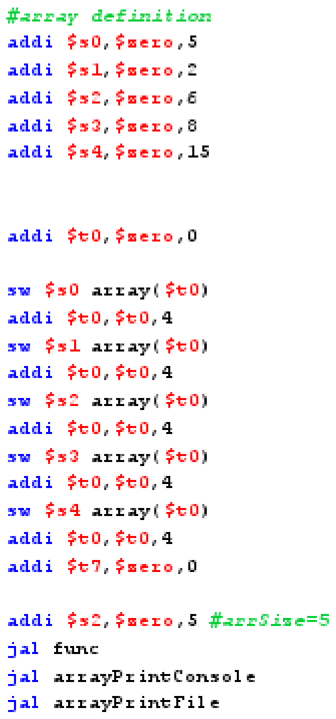


```
#array definition
addi $s0,$zero,3
addi $s1,$zero,10
addi $s2,$zero,7
addi $s3,$zero,9
addi $s4,$zero,4
addi $s5,$zero,11

addi $t0,$zero,0

sw $s0 array($t0)
addi $t0,$t0,4
sw $s1 array($t0)
addi $t0,$t0,4
sw $s2 array($t0)
addi $t0,$t0,4
sw $s3 array($t0)
addi $t0,$t0,4
sw $s4 array($t0)
addi $t0,$t0,4
sw $s5 array($t0)
addi $t0,$t0,4
addi $t7,$zero,0

addi $s2,$zero,6 #arrSize=6
jal func
jal arrayPrintConsole
jal arrayPrintFile
```



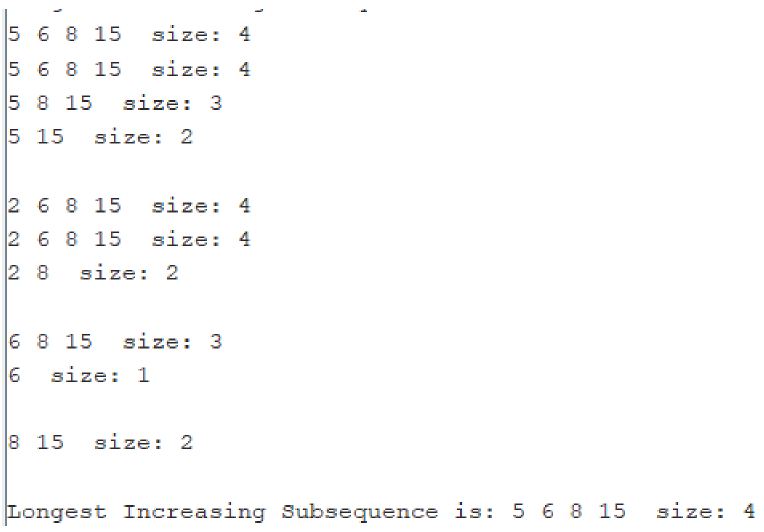
```
3 10 11  size: 3
3 7 9 11  size: 4
3 9 11  size: 3
3 4 11  size: 3
3 11  size: 2

10 11  size: 2
10 11  size: 2
10 11  size: 2
10  size: 1

7 9 11  size: 3
7 9 11  size: 3
7 9  size: 2

9 11  size: 2
9  size: 1

4 11  size: 2

Longest Increasing Subsequence is: 3 7 9 11  size: 4
```

Second Array: {5,2,6,8,15}



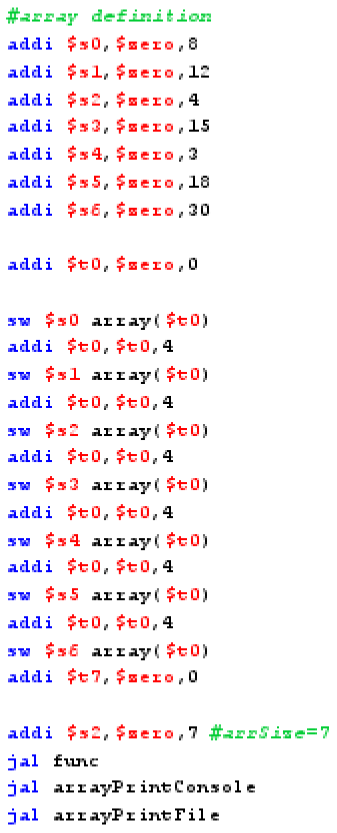
```
#array definition
addi $s0,$zero,5
addi $s1,$zero,2
addi $s2,$zero,6
addi $s3,$zero,8
addi $s4,$zero,15


addi $t0,$zero,0

sw $s0 array($t0)
addi $t0,$t0,4
sw $s1 array($t0)
addi $t0,$t0,4
sw $s2 array($t0)
addi $t0,$t0,4
sw $s3 array($t0)
addi $t0,$t0,4
sw $s4 array($t0)
addi $t0,$t0,4
addi $t7,$zero,0

addi $s2,$zero,5 #arrSize=5
jal func
jal arrayPrintConsole
jal arrayPrintFile
```

Console Output



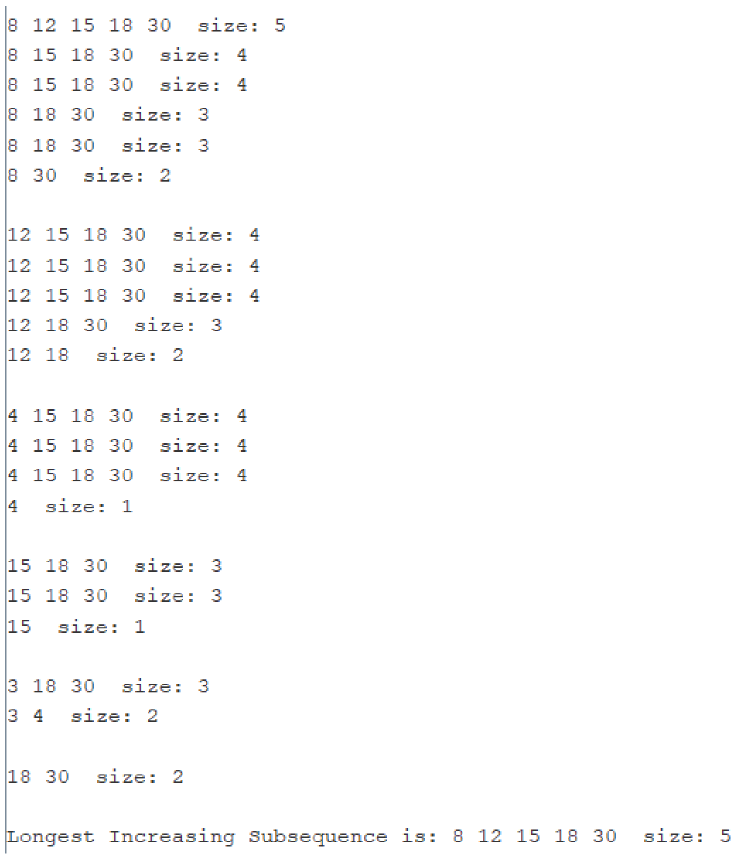
```
5  6  8  15   size:  4
5  6  8  15   size:  4
5  8  15   size:  3
5  15   size:  2

2  6  8  15   size:  4
2  6  8  15   size:  4
2  8   size:  2

6  8  15   size:  3
6   size:  1


8  15   size:  2


Longest  Increasing  Subsequence  is:  5  6  8  15   size:  4
```

Third Array: {8,15,4,15,3,18,30}



```
#array definition
addi $s0,$zero,8
addi $s1,$zero,12
addi $s2,$zero,4
addi $s3,$zero,15
addi $s4,$zero,3
addi $s5,$zero,18
addi $s6,$zero,30

addi $t0,$zero,0

sw $s0 array($t0)
addi $t0,$t0,4
sw $s1 array($t0)
addi $t0,$t0,4
sw $s2 array($t0)
addi $t0,$t0,4
sw $s3 array($t0)
addi $t0,$t0,4
sw $s4 array($t0)
addi $t0,$t0,4
sw $s5 array($t0)
addi $t0,$t0,4
sw $s6 array($t0)
addi $t7,$zero,0

addi $s2,$zero,7 #arrSize=7
jal func
jal arrayPrintConsole
jal arrayPrintFile
```

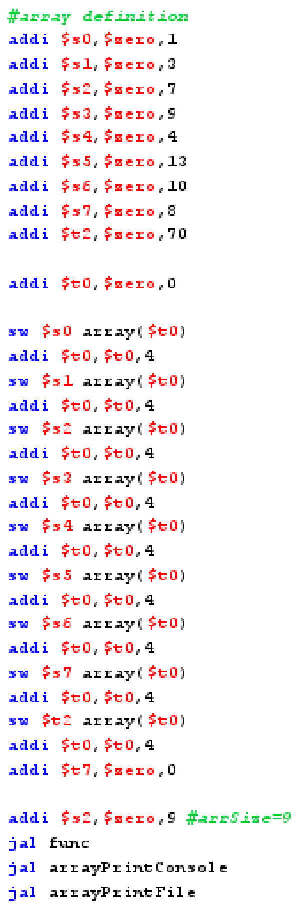
Console Output



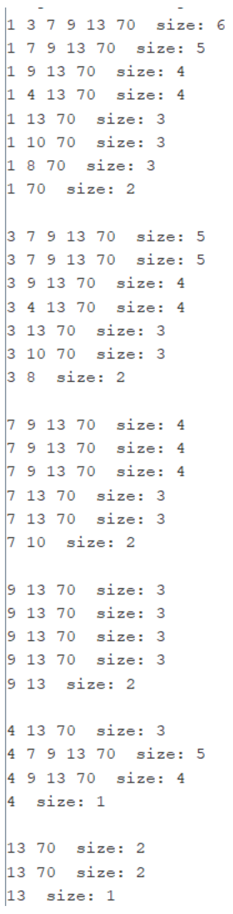
```
8  12  15  18  30   size:  5
8  15  18  30   size:  4
8  15  18  30   size:  4
8  18  30   size:  3
8  18  30   size:  3
8  30   size:  2

12  15  18  30   size:  4
12  15  18  30   size:  4
12  15  18  30   size:  4
12  18  30   size:  3
12  18   size:  2

4  15  18  30   size:  4
4  15  18  30   size:  4
4  15  18  30   size:  4
4   size:  1

15  18  30   size:  3
15  18  30   size:  3
15   size:  1

3  18  30   size:  3
3  4   size:  2

18  30   size:  2

Longest  Increasing  Subsequence  is:  8  12  15  18  30   size:  5
```

Forth Array:{1,3,7,9,4,13,10,8,70}

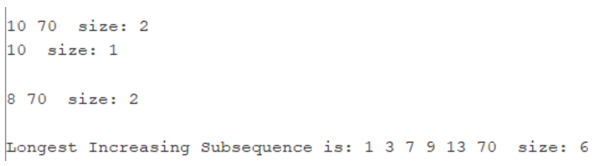
Console Output



```
#array definition
addi $s0,$zero,1
addi $s1,$zero,3
addi $s2,$zero,7
addi $s3,$zero,9
addi $s4,$zero,4
addi $s5,$zero,13
addi $s6,$zero,10
addi $s7,$zero,8
addi $t2,$zero,70

addi $t0,$zero,0

sw $s0 array($t0)
addi $t0,$t0,4
sw $s1 array($t0)
addi $t0,$t0,4
sw $s2 array($t0)
addi $t0,$t0,4
sw $s3 array($t0)
addi $t0,$t0,4
sw $s4 array($t0)
addi $t0,$t0,4
sw $s5 array($t0)
addi $t0,$t0,4
sw $s6 array($t0)
addi $t0,$t0,4
sw $s7 array($t0)
addi $t0,$t0,4
sw $t2 array($t0)
addi $t0,$t0,4
addi $t7,$zero,0

addi $s2,$zero,9 #arrSize=9
jal func
jal arrayPrintConsole
jal arrayPrintFile
```



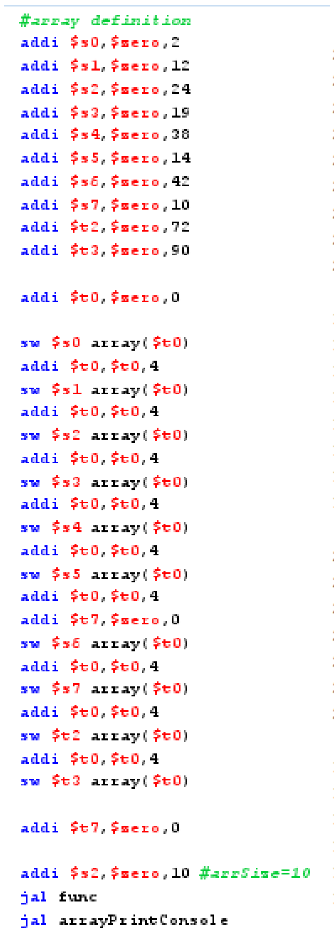
```
1 3 7 9 13 70  size: 6
1 7 9 13 70  size: 5
1 9 13 70  size: 4
1 4 13 70  size: 4
1 13 70  size: 3
1 10 70  size: 3
1 8 70  size: 3
1 70  size: 2

3 7 9 13 70  size: 5
3 7 9 13 70  size: 5
3 9 13 70  size: 4
3 4 13 70  size: 4
3 13 70  size: 3
3 10 70  size: 3
3 8  size: 2

7 9 13 70  size: 4
7 9 13 70  size: 4
7 9 13 70  size: 4
7 13 70  size: 3
7 13 70  size: 3
7 10  size: 2

9 13 70  size: 3
9 13 70  size: 3
9 13 70  size: 3
9 13 70  size: 3
9 13  size: 2

4 13 70  size: 3
4 7 9 13 70  size: 5
4 9 13 70  size: 4
4  size: 1

13 70  size: 2
13 70  size: 2
13  size: 1
```



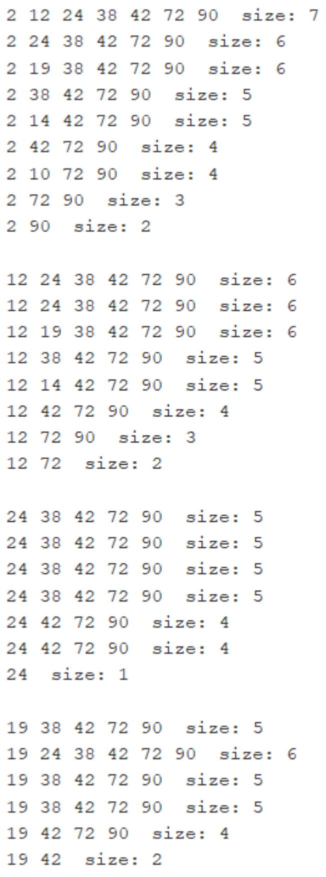
```
10 70  size: 2
10  size: 1

8 70  size: 2

Longest Increasing Subsequence is: 1 3 7 9 13 70  size: 6
```

Fifth Array: {2,12,24,19,38,14,42,10,72,90}

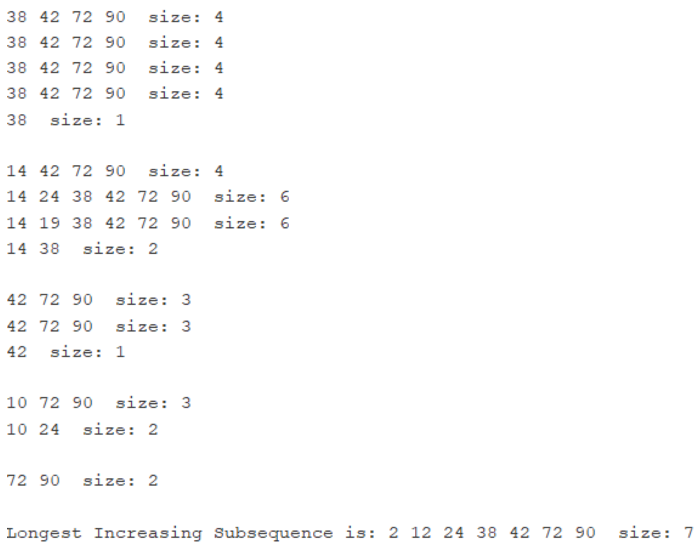
Console Output



```
#array definition
addi $s0,$zero,2
addi $s1,$zero,12
addi $s2,$zero,24
addi $s3,$zero,19
addi $s4,$zero,38
addi $s5,$zero,14
addi $s6,$zero,42
addi $s7,$zero,10
addi $t2,$zero,72
addi $t3,$zero,90

addi $t0,$zero,0

sw $s0 array($t0)
addi $t0,$t0,4
sw $s1 array($t0)
addi $t0,$t0,4
sw $s2 array($t0)
addi $t0,$t0,4
sw $s3 array($t0)
addi $t0,$t0,4
sw $s4 array($t0)
addi $t0,$t0,4
sw $s5 array($t0)
addi $t0,$t0,4
addi $t7,$zero,0
sw $s6 array($t0)
addi $t0,$t0,4
sw $s7 array($t0)
addi $t0,$t0,4
sw $t2 array($t0)
addi $t0,$t0,4
sw $t3 array($t0)

addi $t7,$zero,0

addi $s2,$zero,10 #arrSize=10
jal func
jal arrayPrintConsole
```

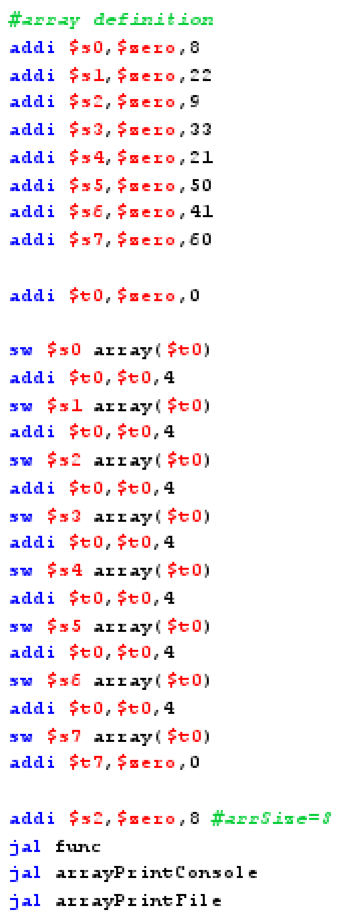


```
2 12 24 38 42 72 90  size: 7
2 24 38 42 72 90  size: 6
2 19 38 42 72 90  size: 6
2 38 42 72 90  size: 5
2 14 42 72 90  size: 5
2 42 72 90  size: 4
2 10 72 90  size: 4
2 72 90  size: 3
2 90  size: 2

12 24 38 42 72 90  size: 6
12 24 38 42 72 90  size: 6
12 19 38 42 72 90  size: 6
12 38 42 72 90  size: 5
12 14 42 72 90  size: 5
12 42 72 90  size: 4
12 72 90  size: 3
12 72  size: 2

24 38 42 72 90  size: 5
24 38 42 72 90  size: 5
24 38 42 72 90  size: 5
24 38 42 72 90  size: 5
24 42 72 90  size: 4
24 42 72 90  size: 4
24  size: 1

19 38 42 72 90  size: 5
19 24 38 42 72 90  size: 6
19 38 42 72 90  size: 5
19 38 42 72 90  size: 5
19 42 72 90  size: 4
19 42  size: 2
```



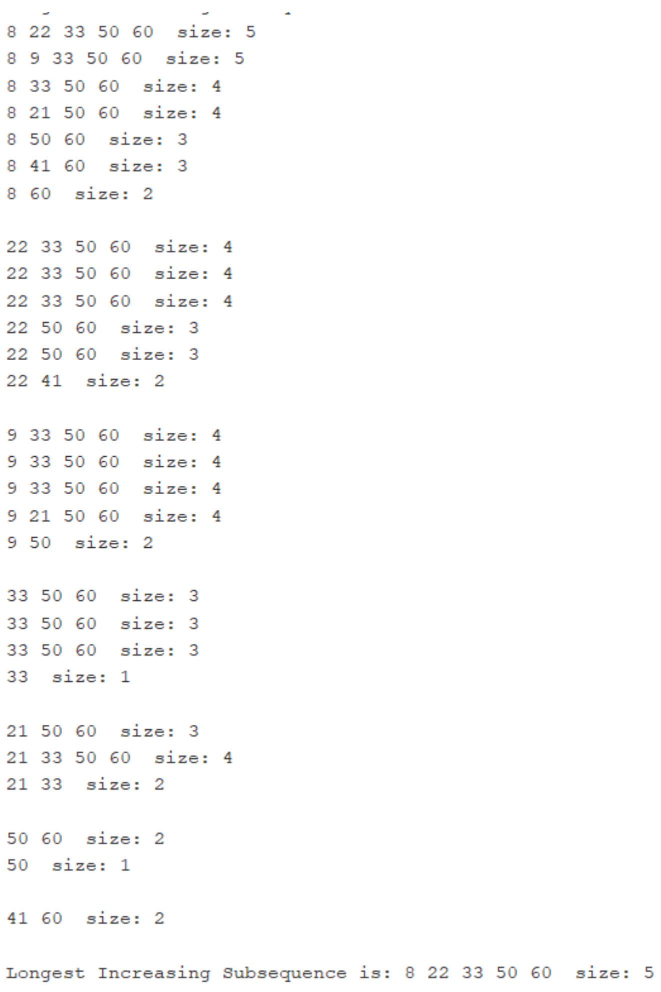
```
38 42 72 90  size: 4
38 42 72 90  size: 4
38 42 72 90  size: 4
38 42 72 90  size: 4
38  size: 1

14 42 72 90  size: 4
14 24 38 42 72 90  size: 6
14 19 38 42 72 90  size: 6
14 38  size: 2

42 72 90  size: 3
42 72 90  size: 3
42  size: 1

10 72 90  size: 3
10 24  size: 2

72 90  size: 2

Longest Increasing Subsequence is: 2 12 24 38 42 72 90  size: 7
```

Sixth Array: {8,22,9,33,21,50,41,60}



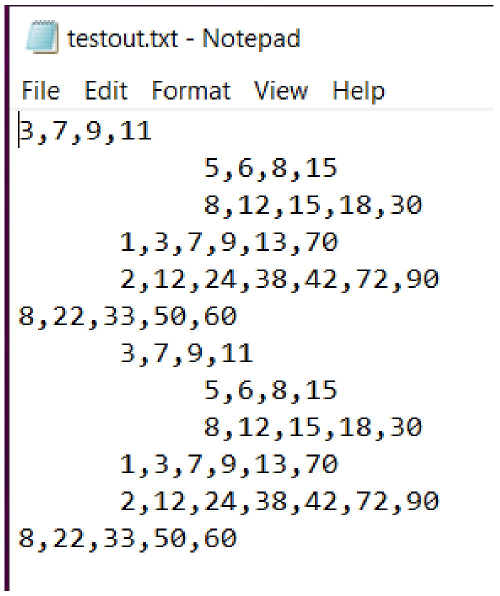
```
#array definition
addi $s0,$zero,8
addi $s1,$zero,22
addi $s2,$zero,9
addi $s3,$zero,33
addi $s4,$zero,21
addi $s5,$zero,50
addi $s6,$zero,41
addi $s7,$zero,60

addi $t0,$zero,0

sw $s0 array($t0)
addi $t0,$t0,4
sw $s1 array($t0)
addi $t0,$t0,4
sw $s2 array($t0)
addi $t0,$t0,4
sw $s3 array($t0)
addi $t0,$t0,4
sw $s4 array($t0)
addi $t0,$t0,4
sw $s5 array($t0)
addi $t0,$t0,4
sw $s6 array($t0)
addi $t0,$t0,4
sw $s7 array($t0)
addi $t7,$zero,0

addi $s2,$zero,8 #arrSize=8
jal func
jal arrayPrintConsole
jal arrayPrintFile
```

Console Output



```
8 22 33 50 60  size: 5
8 9 33 50 60  size: 5
8 33 50 60  size: 4
8 21 50 60  size: 4
8 50 60  size: 3
8 41 60  size: 3
8 60  size: 2

22 33 50 60  size: 4
22 33 50 60  size: 4
22 33 50 60  size: 4
22 50 60  size: 3
22 50 60  size: 3
22 41  size: 2

9 33 50 60  size: 4
9 33 50 60  size: 4
9 33 50 60  size: 4
9 21 50 60  size: 4
9 50  size: 2

33 50 60  size: 3
33 50 60  size: 3
33 50 60  size: 3
33  size: 1

21 50 60  size: 3
21 33 50 60  size: 4
21 33  size: 2

50 60  size: 2
50  size: 1

41 60  size: 2

Longest Increasing Subsequence is: 8 22 33 50 60  size: 5
```

Write to File output(testout.txt)

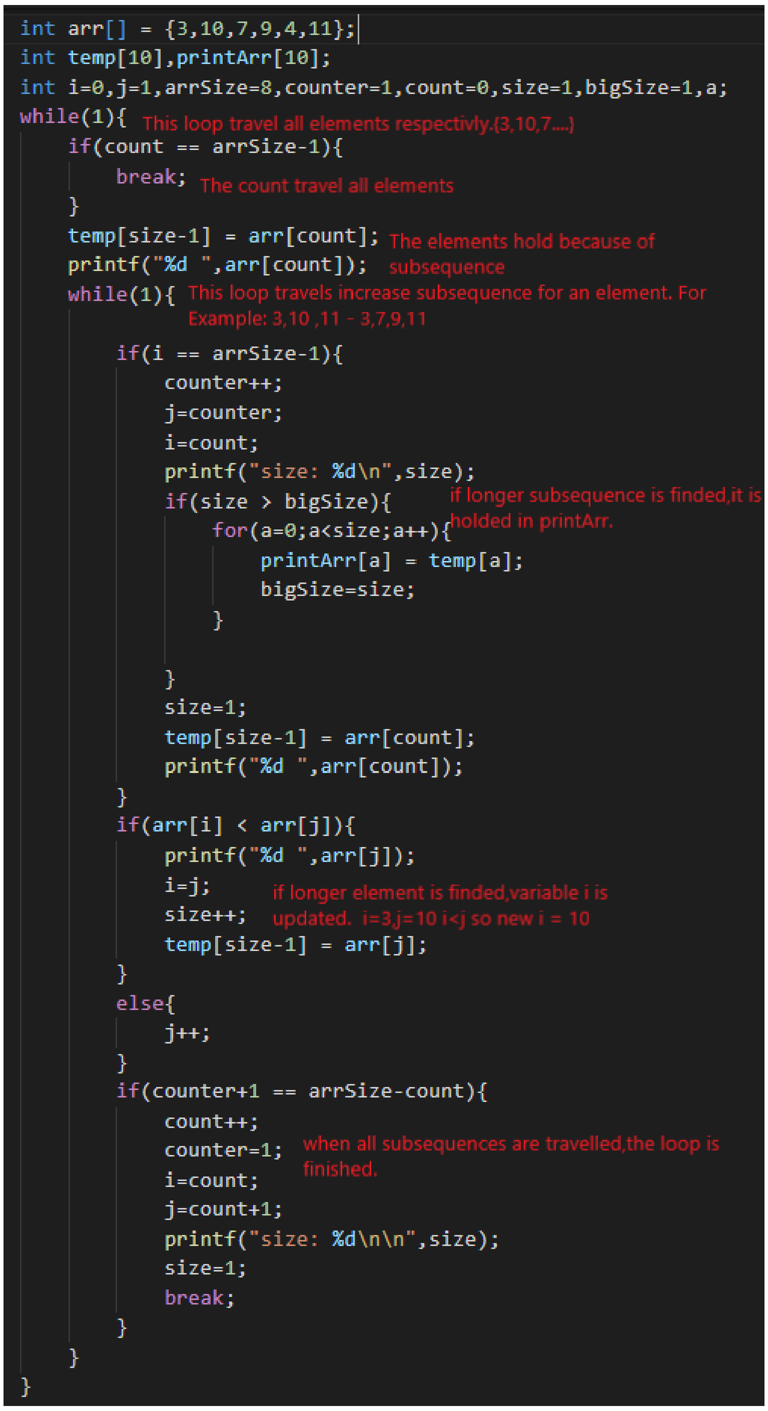


testout.txt - Notepad

File Edit Format View Help

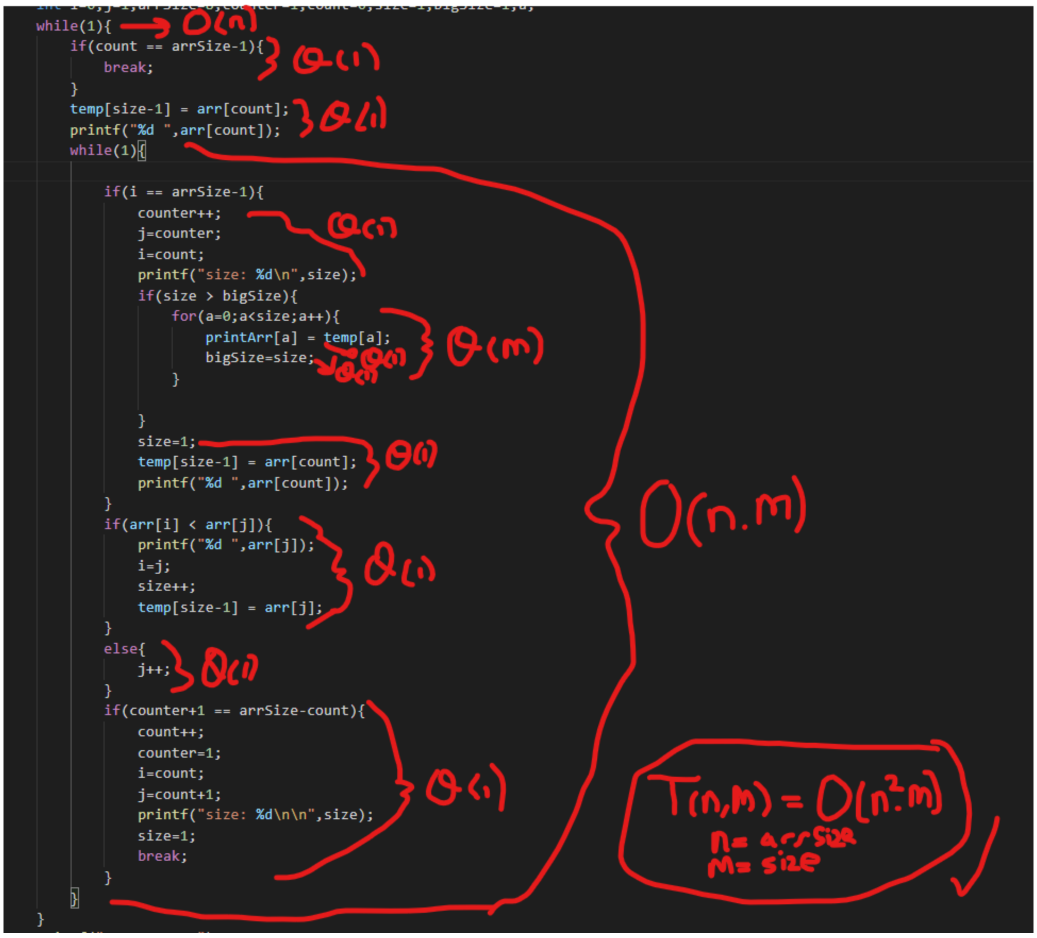
```
3,7,9,11
           5,6,8,15
           8,12,15,18,30
      1,3,7,9,13,70
      2,12,24,38,42,72,90
8,22,33,50,60
      3,7,9,11
           5,6,8,15
           8,12,15,18,30
      1,3,7,9,13,70
      2,12,24,38,42,72,90
8,22,33,50,60
```

3. Explanation of Algortihm



```
int arr[] = {3,10,7,9,4,11};
int temp[10],printArr[10];
int i=0,j=1,arrSize=8,counter=1,count=0,size=1,bigSize=1,a;
while(1){  This loop travel all elements respectivly.{3,10,7....}
    if(count == arrSize-1){
        break;  The count travel all elements
    }
    temp[size-1] = arr[count];  The elements hold because of
    printf("%d ",arr[count]);   subsequence
    while(1){  This loop travels increase subsequence for an element. For
               Example: 3,10 ,11 - 3,7,9,11
        if(i == arrSize-1){
            counter++;
            j=counter;
            i=count;
            printf("size: %d\n",size);
            if(size > bigSize){         if longer subsequence is finded,it is
                for(a=0;a<size;a++){    holded in printArr.
                    printArr[a] = temp[a];
                    bigSize=size;
                }

            }
            size=1;
            temp[size-1] = arr[count];
            printf("%d ",arr[count]);
        }
        if(arr[i] < arr[j]){
            printf("%d ",arr[j]);
            i=j;       if longer element is finded,variable i is
            size++;    updated.  i=3,j=10 i<j so new i = 10
            temp[size-1] = arr[j];
        }
        else{
            j++;
        }
        if(counter+1 == arrSize-count){
            count++;
            counter=1;  when all subsequences are travelled,the loop is
            i=count;    finished.
            j=count+1;
            printf("size: %d\n\n",size);
            size=1;
            break;
        }
    }
}
```

4. Space Complexity of Algorithm



```c
int i=0,j=1,arrSize=0,counter=1,count=0,size=1,bigSize=1,a;
while(1){
    if(count == arrSize-1){
        break;
    }
    temp[size-1] = arr[count];
    printf("%d ",arr[count]);
    while(1){

        if(i == arrSize-1){
            counter++;
            j=counter;
            i=count;
            printf("size: %d\n",size);
            if(size > bigSize){
                for(a=0;a<size;a++){
                    printArr[a] = temp[a];
                    bigSize=size;
                }

            }
            size=1;
            temp[size-1] = arr[count];
            printf("%d ",arr[count]);
        }
        if(arr[i] < arr[j]){
            printf("%d ",arr[j]);
            i=j;
            size++;
            temp[size-1] = arr[j];
        }
        else{
            j++;
        }
        if(counter+1 == arrSize-count){
            count++;
            counter=1;
            i=count;
            j=count+1;
            printf("size: %d\n\n",size);
            size=1;
            break;
        }
    }
}
```