1) Pseudocode and analysis

```
int array[] = { 3, 10, 7, 9, 4, 11}
int result[] = {}
int temp[] = {}
int arraySize = 6
int tempIndex = 0
int maxSize = 0
```

n = array size
m = temp size
p = result size

```
main():                                              Θ(n²+m+p)
        calcLongest(arr,0,temp)  Θ(n²+m)
        printResult(result)  Θ(p)
--------------------------------------------------------------------------------
calcLongest(int array[],int arrIndex,int tempArr[]):
        if(arrIndex >= arr.length):  Θ(1)
                if(tempIndex > maxSize):  Θ(1)
                        maxSize = tempIndex  Θ(1)
                        result = temp  Θ(1)
                printCandidate(temp)  Θ(m)
                return  Θ(1)

        if(temp.size() == 0 || arr[arrIndex] > temp[tempIndex -1])  Θ(1)
        temp[tempIndex-1] = arr[arrIndex]  Θ(1)
                tempIndex++  Θ(1)
                calcLongest(arr,arrIndex+1,temp)
                tempIndex-1 #remove the last element of the temp  Θ(1)

        calculateLongest(arr,arrIndex+1,temp)

--------------------------------------------------------------------------------
printCandidade(int tempArr[]):
        print("A candidade increasing subsequence is = ")  Θ(1)
        for(int i=0;I<tempIndex,i++)  Θ(1)
                print(tempArr[I] + ',')  Θ(1)
        print("size: " + tempIndex)  Θ(1)
--------------------------------------------------------------------------------
printResult(int resultArr[]):
        print("The longest increasing subsequence is = ")  Θ(1)
        for(int i=0;I<arrSize,i++)  Θ(1)
                print(resultArr[I] + ',')  Θ(1)
        print("size: " +arrSize)  Θ(1)
```

Base Case: $\Theta(m)$

$T(0) = \Theta(m)$

$T(n) = T(n-1)^2 + \Theta(1)$
$= T(n-2)^2 + 4 \cdot \Theta(1)$
$\vdots$
$= T(0) + 2^n \cdot \Theta(1)$
$= \Theta(2^n) + \Theta(m)$ (calcLongest)

printCandidade: $T(m) = \Theta(m)$

printResult: $T(p) = \Theta(p)$

Time Complexity

n = arr size, m = temp size, p = result size

PrintTemp: $\Theta(m)$ PrintResult: $\Theta(p)$

calcLongest: $O(n^2 + m)$ → Because of the print statement in base case

main: $O(n^2 + m + p)$

Explanation of the calcLongest subroutine:

Base case: if the arrIndex is larger then arr length it means whole array is traveled. If temp array is larger then current LIS(result array), change them, print the candidate and finish the process.

If base case is not the situation: If temp array is empty or current array element is larger than the previous temp element, add the next element of the array to the temp and recursively repeat the process and remove the added element from temp after returning. Without any conditions, recursively repeat the process without adding anything to temp. This way, all the possible subsequences are considered.

Example:

A candidade increasing subsequence is = 10,22,33,50, Length = 4 **#next element (50) is added to temp.**

A candidade increasing subsequence is = 10,22,33,41,60,80, Length = 6 **#next element (50) is not added to temp and found a longer subsequence.**

2) Test Cases

a)

```
Array is = 3,10,7,9,4,11,  Length = 6
A candidade increasing subsequence is = 3,10,11,  Length = 3
A candidade increasing subsequence is = 3,10,  Length = 2
A candidade increasing subsequence is = 3,7,9,11,  Length = 4
A candidade increasing subsequence is = 3,7,9,  Length = 3
A candidade increasing subsequence is = 3,7,11,  Length = 3
A candidade increasing subsequence is = 3,7,  Length = 2
A candidade increasing subsequence is = 3,9,11,  Length = 3
A candidade increasing subsequence is = 3,9,  Length = 2
A candidade increasing subsequence is = 3,4,11,  Length = 3
A candidade increasing subsequence is = 3,4,  Length = 2
A candidade increasing subsequence is = 3,11,  Length = 2
A candidade increasing subsequence is = 3,  Length = 1
A candidade increasing subsequence is = 10,11,  Length = 2
A candidade increasing subsequence is = 10,  Length = 1
A candidade increasing subsequence is = 7,9,11,  Length = 3
A candidade increasing subsequence is = 7,9,  Length = 2
A candidade increasing subsequence is = 7,11,  Length = 2
A candidade increasing subsequence is = 7,  Length = 1
A candidade increasing subsequence is = 9,11,  Length = 2
A candidade increasing subsequence is = 9,  Length = 1
A candidade increasing subsequence is = 4,11,  Length = 2
A candidade increasing subsequence is = 4,  Length = 1
A candidade increasing subsequence is = 11,  Length = 1
The longest increasing subsequence is = 3,7,9,11,  Length = 4
--------------------------------------
```

b)

```
Array is = 10,22,9,33,21,50,41,60,80,  Length = 9
A candidade increasing subsequence is = 10,22,33,50,60,80,  Length = 6
A candidade increasing subsequence is = 10,22,33,50,60,  Length = 5
A candidade increasing subsequence is = 10,22,33,50,80,  Length = 5
A candidade increasing subsequence is = 10,22,33,50,  Length = 4
A candidade increasing subsequence is = 10,22,33,41,60,80,  Length = 6
A candidade increasing subsequence is = 10,22,33,41,60,  Length = 5
A candidade increasing subsequence is = 10,22,33,41,80,  Length = 5
A candidade increasing subsequence is = 10,22,33,41,  Length = 4
A candidade increasing subsequence is = 10,22,33,60,80,  Length = 5
A candidade increasing subsequence is = 10,22,33,60,  Length = 4
A candidade increasing subsequence is = 10,22,33,80,  Length = 4
A candidade increasing subsequence is = 10,22,33,  Length = 3
A candidade increasing subsequence is = 10,22,50,60,80,  Length = 5
A candidade increasing subsequence is = 10,22,50,60,  Length = 4
A candidade increasing subsequence is = 10,22,50,80,  Length = 4
A candidade increasing subsequence is = 10,22,50,  Length = 3
A candidade increasing subsequence is = 10,22,41,60,80,  Length = 5
A candidade increasing subsequence is = 10,22,41,60,  Length = 4
A candidade increasing subsequence is = 10,22,41,80,  Length = 4
A candidade increasing subsequence is = 10,22,41,  Length = 3
A candidade increasing subsequence is = 10,22,60,80,  Length = 4
A candidade increasing subsequence is = 10,22,60,  Length = 3
A candidade increasing subsequence is = 10,22,80,  Length = 3
A candidade increasing subsequence is = 10,22,  Length = 2
A candidade increasing subsequence is = 10,33,50,60,80,  Length = 5
A candidade increasing subsequence is = 10,33,50,60,  Length = 4
A candidade increasing subsequence is = 10,33,50,80,  Length = 4
```

...

```
A candidade increasing subsequence is = 33,80,  Length = 2
A candidade increasing subsequence is = 33,  Length = 1
A candidade increasing subsequence is = 21,50,60,80,  Length = 4
A candidade increasing subsequence is = 21,50,60,  Length = 3
A candidade increasing subsequence is = 21,50,80,  Length = 3
A candidade increasing subsequence is = 21,50,  Length = 2
A candidade increasing subsequence is = 21,41,60,80,  Length = 4
A candidade increasing subsequence is = 21,41,60,  Length = 3
A candidade increasing subsequence is = 21,41,80,  Length = 3
A candidade increasing subsequence is = 21,41,  Length = 2
A candidade increasing subsequence is = 21,60,80,  Length = 3
A candidade increasing subsequence is = 21,60,  Length = 2
A candidade increasing subsequence is = 21,80,  Length = 2
A candidade increasing subsequence is = 21,  Length = 1
A candidade increasing subsequence is = 50,60,80,  Length = 3
A candidade increasing subsequence is = 50,60,  Length = 2
A candidade increasing subsequence is = 50,80,  Length = 2
A candidade increasing subsequence is = 50,  Length = 1
A candidade increasing subsequence is = 41,60,80,  Length = 3
A candidade increasing subsequence is = 41,60,  Length = 2
A candidade increasing subsequence is = 41,80,  Length = 2
A candidade increasing subsequence is = 41,  Length = 1
A candidade increasing subsequence is = 60,80,  Length = 2
A candidade increasing subsequence is = 60,  Length = 1
A candidade increasing subsequence is = 80,  Length = 1
The longest increasing subsequence is = 10,22,33,41,60,80,  Length = 6
```

c)

```
Array is = 3,10,2,1,20,  Length = 5
A candidade increasing subsequence is = 3,10,20,  Length = 3
A candidade increasing subsequence is = 3,10,  Length = 2
A candidade increasing subsequence is = 3,20,  Length = 2
A candidade increasing subsequence is = 3,  Length = 1
A candidade increasing subsequence is = 10,20,  Length = 2
A candidade increasing subsequence is = 10,  Length = 1
A candidade increasing subsequence is = 2,20,  Length = 2
A candidade increasing subsequence is = 2,  Length = 1
A candidade increasing subsequence is = 1,20,  Length = 2
A candidade increasing subsequence is = 1,  Length = 1
A candidade increasing subsequence is = 20,  Length = 1
The longest increasing subsequence is = 3,10,20,  Length = 3
```

d)

```
Array is = 7,10,1,11,3,12,  Length = 6
A candidade increasing subsequence is = 7,10,11,12,  Length = 4
A candidade increasing subsequence is = 7,10,11,  Length = 3
A candidade increasing subsequence is = 7,10,12,  Length = 3
A candidade increasing subsequence is = 7,10,  Length = 2
A candidade increasing subsequence is = 7,11,12,  Length = 3
A candidade increasing subsequence is = 7,11,  Length = 2
A candidade increasing subsequence is = 7,12,  Length = 2
A candidade increasing subsequence is = 7,  Length = 1
A candidade increasing subsequence is = 10,11,12,  Length = 3
A candidade increasing subsequence is = 10,11,  Length = 2
A candidade increasing subsequence is = 10,12,  Length = 2
A candidade increasing subsequence is = 10,  Length = 1
A candidade increasing subsequence is = 1,11,12,  Length = 3
A candidade increasing subsequence is = 1,11,  Length = 2
A candidade increasing subsequence is = 1,3,12,  Length = 3
A candidade increasing subsequence is = 1,3,  Length = 2
A candidade increasing subsequence is = 1,12,  Length = 2
A candidade increasing subsequence is = 1,  Length = 1
A candidade increasing subsequence is = 11,12,  Length = 2
A candidade increasing subsequence is = 11,  Length = 1
A candidade increasing subsequence is = 3,12,  Length = 2
A candidade increasing subsequence is = 3,  Length = 1
A candidade increasing subsequence is = 12,  Length = 1
The longest increasing subsequence is = 7,10,11,12,  Length = 4
```

e)

```
Array is = 1,2,3,5,4,  Length = 5
A candidade increasing subsequence is = 1,2,3,5,  Length = 4
A candidade increasing subsequence is = 1,2,3,4,  Length = 4
A candidade increasing subsequence is = 1,2,3,  Length = 3
A candidade increasing subsequence is = 1,2,5,  Length = 3
A candidade increasing subsequence is = 1,2,4,  Length = 3
A candidade increasing subsequence is = 1,2,  Length = 2
A candidade increasing subsequence is = 1,3,5,  Length = 3
A candidade increasing subsequence is = 1,3,4,  Length = 3
A candidade increasing subsequence is = 1,3,  Length = 2
A candidade increasing subsequence is = 1,5,  Length = 2
A candidade increasing subsequence is = 1,4,  Length = 2
A candidade increasing subsequence is = 1,  Length = 1
A candidade increasing subsequence is = 2,3,5,  Length = 3
A candidade increasing subsequence is = 2,3,4,  Length = 3
A candidade increasing subsequence is = 2,3,  Length = 2
A candidade increasing subsequence is = 2,5,  Length = 2
A candidade increasing subsequence is = 2,4,  Length = 2
A candidade increasing subsequence is = 2,  Length = 1
A candidade increasing subsequence is = 3,5,  Length = 2
A candidade increasing subsequence is = 3,4,  Length = 2
A candidade increasing subsequence is = 3,  Length = 1
A candidade increasing subsequence is = 5,  Length = 1
A candidade increasing subsequence is = 4,  Length = 1
The longest increasing subsequence is = 1,2,3,4,  Length = 4
```

f)

```
Array is = 90,20,0,31,5,51,6,  Length = 7
A candidade increasing subsequence is = 90,  Length = 1
A candidade increasing subsequence is = 20,31,51,  Length = 3
A candidade increasing subsequence is = 20,31,  Length = 2
A candidade increasing subsequence is = 20,51,  Length = 2
A candidade increasing subsequence is = 20,  Length = 1
A candidade increasing subsequence is = 0,31,51,  Length = 3
A candidade increasing subsequence is = 0,31,  Length = 2
A candidade increasing subsequence is = 0,5,51,  Length = 3
A candidade increasing subsequence is = 0,5,6,  Length = 3
A candidade increasing subsequence is = 0,5,  Length = 2
A candidade increasing subsequence is = 0,51,  Length = 2
A candidade increasing subsequence is = 0,6,  Length = 2
A candidade increasing subsequence is = 0,  Length = 1
A candidade increasing subsequence is = 31,51,  Length = 2
A candidade increasing subsequence is = 31,  Length = 1
A candidade increasing subsequence is = 5,51,  Length = 2
A candidade increasing subsequence is = 5,6,  Length = 2
A candidade increasing subsequence is = 5,  Length = 1
A candidade increasing subsequence is = 51,  Length = 1
A candidade increasing subsequence is = 6,  Length = 1
The longest increasing subsequence is = 0,5,6,  Length = 3
-- program is finished running --
```

3) Missing Parts

I couldn't implement the read and write to file parts. So I used a dynamic array (.space) and changed the array content 6 times and called the calcLongest subroutine 6 times at the main to test the algorithm with 6 different arrays.