Fall 2021 CSE 331/503

HW1 REPORT

Uğur Er

1901042262

1. Function Descriptions

1. readfile: reads file
2. readOneArrayMain: it gets 1 line from the file. If the end of the file is reached, it starts the writing process.
3. takeOneArrFromFile: Reads character by character from the file and assigns it to $a0. And called by readOneArrayMain.
4. myAtoi: Converts a string number to an integer.Pseudo Code:

```
void myAtoi(int arr[],int n){
        int arr[10];
        int k=0;
        int res=0;
        int i=0;
        while(str[i]!='\0'){
            if(str[i]!=','){
                res = res * 10 + str[i] - '0';
            }
            else{
                arr[k]=res;
                res=0;
                k++;
                i++;
            }
            i++;
            if(str[i]=='\0'){
                arr[k]=res;
                k++;
                break;
            }
        }
        for(i=0;i<k;i++){
            printf("%d ",arr[i]);
        }
    }
```

5. lis: pseudo code(C code)

```c
int lis(int arr[],int n){
    int sizeOfLis[n];
    int answerIndex[n];
    int maxIndex=1; //longest increasing subsequence size index
    in sizeOf Lis
    for(int i=0; i < n; i++){
        sizeOfLis[i] = 1;
        answerIndex[i] = -1;
    }

    for(int i=1; i < n; i++){
        for(int j=0; j < i; j++){
            if(arr[i] > arr[j]){
                if(sizeOfLis[j] + 1 > sizeOfLis[i]){
                    sizeOfLis[i] = sizeOfLis[j] + 1;
                    if(sizeOfLis[i] > sizeOfLis[maxIndex]){
                        maxIndex = i;
                    }
                    if(answerIndex[sizeOfLis[i]-2]==-1){
                        answerIndex[sizeOfLis[i]-2] = j;
                    }
                }
            }
            if(arr[i]<arr[j]&&sizeOfLis[i]==sizeOfLis[j]){
                if(i!=n-1)
                    sizeOfLis[j]-=1;
            }

        }
    }
    answerIndex[sizeOfLis[maxIndex]-1]=maxIndex;
```

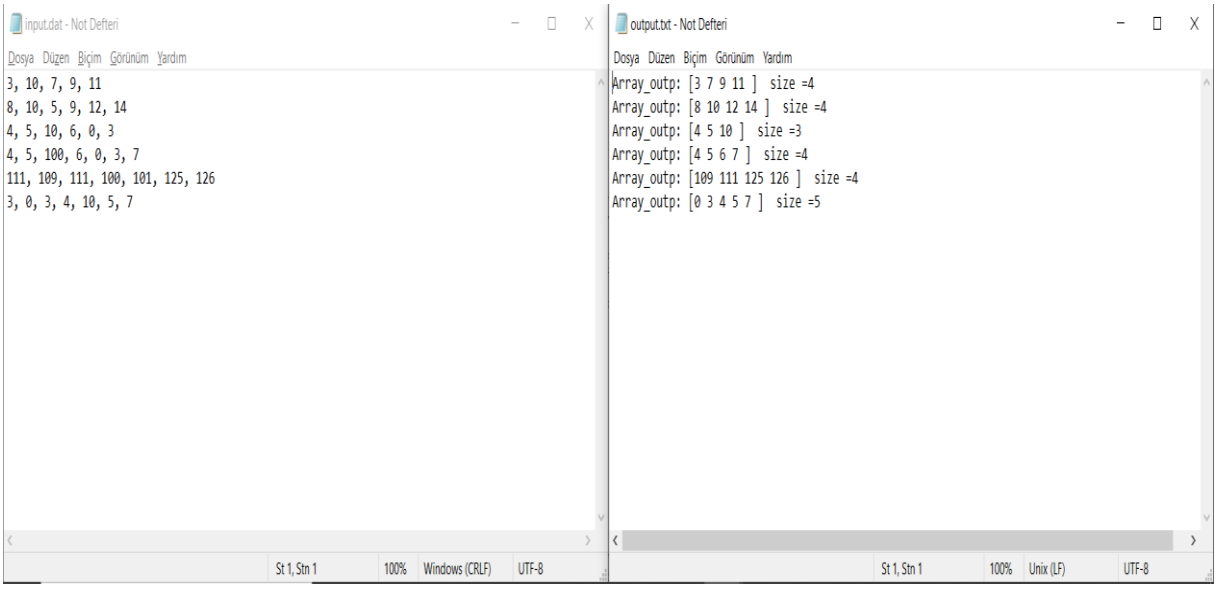
```c
for(int i=0;i<sizeOfLis[maxIndex];i++)
    printf("%d ",arr[answerIndex[i]]);
printf("\n");
```

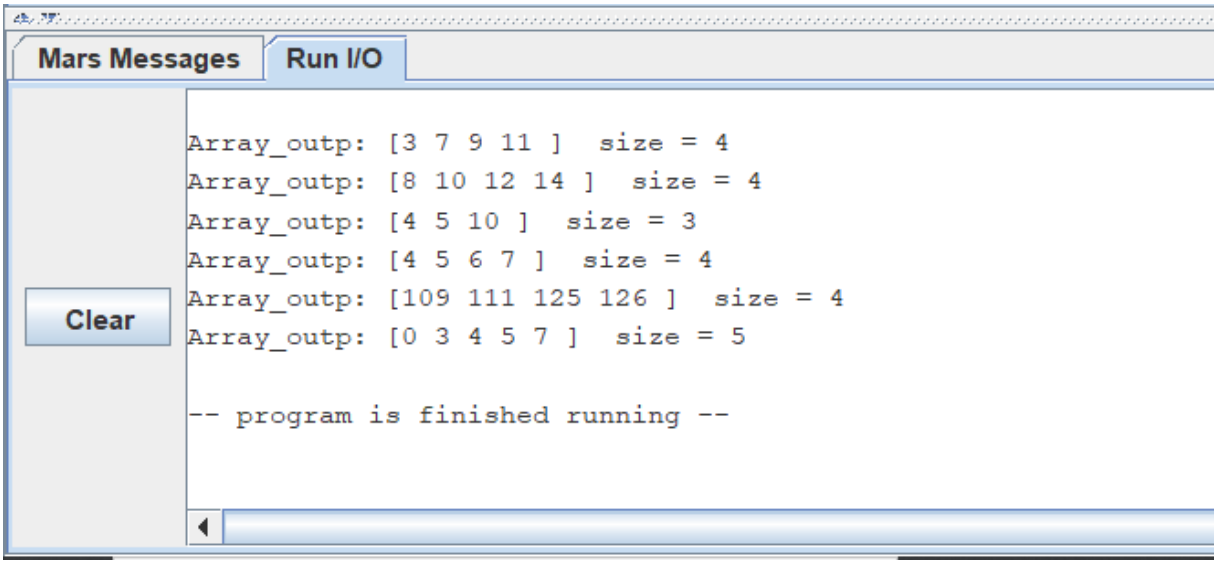
this is the c code of the longest increasing subsequence. Starting from the 1st index of the array, if the first index is greater than 0, increase the index in the sizeOfLis array by 1, and if the answer array has -1 (ie if it is empty) it inserts the index of smaller number in the answerIndex array. answerIndex array keeps the indexes of the elements of the longest increasing subset.
The second if in the code, if the lis value of the number 2 is the same and the larger number is to the left of the small number, it

reduces the lis value of the larger number by 1, allowing the search for the answer to continue with the smaller number.

6. printLongestSub: it prints longest increasing subsequence to the the terminal and to the file. Then it returns to the beginning again to the readArrayMain function.
7. writeToFile: Opens the file and writes the writeText string.
8. writeSizeText: it just write “]size: “character by character
9. writeOutputText: it just write “Arr_output: [” character by character

2.Test Cases:





I tried to indicate all possible test possibilities in 6 different inputs in the input text.

3. Some Calculations:

1. Time Complexity of longest increasing subsequence algorithm: There are 2 for in the code. The operations inside the for and all the operations outside take constant time. So T(n) is O(n^2)
2. Space Complexity of longest increasing subsequence algorithm: There are 2 n-element arrays in the function. So the space Complexity is O(n)