Report File

Homework 2

Md sarwar Hossain

Id: 161044121

To find the longest subsequence in an array , I wrote a code in c then converted the code into MIPS instruction set. I managed to find the length of the longest sequence which will be displayed in terminal.

However I couldn't print the sequence itself because of recursion. But I implemented the recursion to find the sequence in C program. I found the sub-sequences and their parent index which are stored in array in mips program. Using recursive approach in this array would give the sequence.

In finding the longest sequence length, the time complexity is O(n^2) as there are two loop and each going up to the length n.

I used one single array which will give me a space complexity of O(n).

As a pseudocode, I am presenting the c program. However it doesn't include the part of reading from file. I used static file path instead of relative file path.

The program will give a error if the correct file path is not used. So in order to test the program, it should be given with it's static path to the input.txt file.

Instead of brute forcing the whole array, I used dynamic programming solution to solve the problem which is much more optimized than brute forcing.

```c
#include <stdio.h>

void parent(int array[6], int index, int mainArray[6])
{
    if (index == 0)
    {
        return;
    }
```

```
    index = array[index];
    parent(array, index, mainArray);

    printf("%d ", mainArray[index]);
}
int main()
{
    //int array[] = {0, 4, 12, 2, 10, 6, 9, 13, 3, 11, 7,
15};
    int array[] = {3, 10, 7, 9, 4, 11};
    int length = sizeof(array) / sizeof(array[0]);
    int seq[12];
    int subseq[12];
    for (int i = 0; i < length; i++)
    {
        subseq[i] = -1;
    }

    for (int i = 0; i < length; i++)
    {
        seq[i] = 1;
    }
    for (int j = 1; j < length; j++)
    {
        for (int i = 0; i < j; i++)
        {
            if (array[i] < array[j])
            {
                int temp = seq[i];
                temp += 1;
                if (seq[j] < temp)
                {
                    seq[j] = temp;
                }
                if (seq[j] <= temp)
                {

                    subseq[j] = i;
```

```
                }
            }
        }
    }
    printf("sequence array \n");
    for (int i = 0; i < length; i++)
    {
        printf("%d ", seq[i]);
    }
    printf("\n");
    printf("sub sequence array \n");
    for (int i = 0; i < length; i++)
    {
        printf("%d ", subseq[i]);
    }
    printf("\n");
    int highestIndex = 0;
    for (int i = 1; i < length; i++)
    {
        if (seq[i] > seq[highestIndex])
        {
            highestIndex = i;
        }
    }
    printf("highest index : %d\n", highestIndex);
    parent(subseq, highestIndex, array);
    printf("%d ", array[highestIndex]);
    printf("\n----------------\n");
    printf("%d ", array[highestIndex]);
    while (highestIndex != 0)
    {
        highestIndex = subseq[highestIndex];
        printf("%d ", array[highestIndex]);
    }
}
```