Problem 1. [30 points]
In class, we discussed a dynamic program to solve the weighted interval scheduling problem. Use the algorithm presented in class, which can also be found in page 258, to find optimal schedule for the following jobs:

- Job 1: $s_1 = 1$, $f_1 = 5$, $v_1 = 20$
- Job 2: $s_1 = 4$, $f_1 = 7$, $v_1 = 10$
- Job 3: $s_1 = 6$, $f_1 = 9$, $v_1 = 5$
- Job 4: $s_1 = 9$, $f_1 = 15$, $v_1 = 15$
- Job 5: $s_1 = 12$, $f_1 = 16$, $v_1 = 10$

Show the vector $M$ as generated by the algorithm. Then,

1. Describe the order in which you entered values into the vector.
2. What does each entry represent?
3. The output of this algorithm is the value of the best schedule of compatible jobs. What is the schedule itself?
Problem 2. [20 points]

KT Chapter 6 - Problem 1. Let \( G = (V, E) \) be an undirected graph with \( n \) nodes. A subset of the nodes is called an independent set if no two of them are joined by an edge. Finding large independent sets is difficult in general; but here we will see that it can be done efficiently if the graph is simple enough.

Call a graph \( G = (V, E) \) a path if its nodes can be written as \( v_1, v_2, ..., v_n \), with an edge between \( v_i \) and \( v_j \) if and only if the numbers \( i \) and \( j \) differ by exactly 1. With each node \( v_i \), we associate a positive integer weight \( w_i \).

Consider, for example, the five-node path drawn in Figure 6.28. The weights are the numbers drawn inside the nodes. The goal in this question is to solve the following problem: Find an independent set in a path \( G \) whose total weight is as large as possible.

(a) Give an example to show that the following algorithm does not always find an independent set of maximum total weight.

The "heaviest-first" greedy algorithm

```
Start with \( S \) equal to the empty set
While some node remains in \( G \)
    Pick a node \( v_i \) of maximum weight
    Add \( v_i \) to \( S \)
    Delete \( v_i \) and its neighbors from \( G \)
Endwhile
Return \( S \)
```

(b) Give an example to show that the following algorithm also does not always find an independent set of maximum total weight.

```
Let \( S_1 \) be the set of all \( v_i \) where \( i \) is an odd number
Let \( S_2 \) be the set of all \( v_i \) where \( i \) is an even number
(Note that \( S_1 \) and \( S_2 \) are both independent sets)
Determine which of \( S_1 \) or \( S_2 \) has greater total weight,
and return this one
```

(c) Give an algorithm that takes an \( n \)-node path \( G \) with weights and returns an independent set of maximum total weight. Analyze the algorithm. The running time should be polynomial in \( n \), independent of the values of the weights.
Problem 3.  [20 points]
Given two strings $str_1$ and $str_2$ and the below edit operations that can performed on $str_1$, which are all of equal cost ($c$). Find minimum number of edits (operations) required to convert $str_1$ into $str_2$.

- Insert
- Remove
- Replace

Examples:

Input: $str_1 = \text{“geek”}$, $str_2 = \text{”greek”}$ Output: 1
We can convert $str_1$ into $str_2$ by inserting an ‘r’.

Input: $str_1 = \text{“cat”}$, $str_2 = \text{“cut”}$ Output: 1
We can convert $str_1$ into $str_2$ by replacing ‘a’ with ‘u’.

Input: $str_1 = \text{“sunday”}$, $str_2 = \text{“saturday”}$ Output: 3
Last three and first characters are same. We basically need to convert “un” to “atu”. This can be done by replacing ‘n’ with ‘r’, inserting t, inserting a

1. Write the recurrence that solves this problem.
2. Write the pseudocode of a dynamic program to solve it.