# CS590 homework 4 – Dynamic Programming, Greedy Algorithms

**Name:- Harshal Vaidya**
**CWID: 10468204**

**Q4. Find the maximum alignment for X = dcdcbacbbb and Y = acdccabdbb by using the Smith-Waterman algorithm(see slides). Execute the pseudocode algorithm and fill the necessary tables H and P in a bottom-up fashion. Reconstruct the strings X‘ and Y‘ using the tables H and P.**

**Table for H:**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | -1 | -1 | 2 | 1 | 0 | -1 | -1 | 2 | 1 | 0 |
| 0 | -1 | 1 | 1 | 4 | 3 | 2 | 1 | 1 | 1 | 0 |
| 0 | -1 | 0 | 3 | 3 | 3 | 2 | 1 | 3 | 2 | 1 |
| 0 | -1 | 1 | 2 | 5 | 5 | 4 | 3 | 2 | 2 | 1 |
| 0 | -1 | 0 | 1 | 4 | 4 | 4 | 6 | 5 | 4 | 4 |
| 0 | 2 | 1 | 0 | 3 | 3 | 6 | 5 | 5 | 4 | 3 |
| 0 | 1 | 4 | 3 | 2 | 5 | 5 | 5 | 4 | 4 | 3 |
| 0 | 0 | 3 | 3 | 2 | 4 | 4 | 7 | 6 | 6 | 6 |
| 0 | -1 | 2 | 2 | 2 | 3 | 3 | 6 | 6 | 8 | 8 |
| 0 | -1 | 1 | 1 | 1 | 2 | 2 | 5 | 5 | 8 | 10 |

**Table for P:**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| - | d | d | d | l | L | D | d | d | l | l |
| - | d | d | u | d | D | L | l | u | d | d |
| - | d | u | d | u | D | D | d | d | l | l |
| - | d | d | u | d | D | L | l | u | d | d |
| - | d | u | u | u | D | D | D | l | d | d |
| - | d | l | u | u | D | D | u | d | d | d |
| - | u | d | l | d | D | U | d | d | d | d |
| - | u | u | d | d | U | D | d | l | d | d |
| - | d | u | d | D | U | D | d | d | d | d |
| - | d | u | d | D | U | D | d | d | d | d |

**Exercise 15.1-2:**

**Show, by means of a counter example, that the following "greedy" strategy does not always determine an optimal way to cut rods. Define the density of a rod of length i to be pi/i, that is, its value per inch. The greedy strategy for a rod of length n cuts off a first piece of length i, where 1 <= i <= n, having maximum density. It then continues by applying the greedy strategy to the remaining piece of length n – i.**

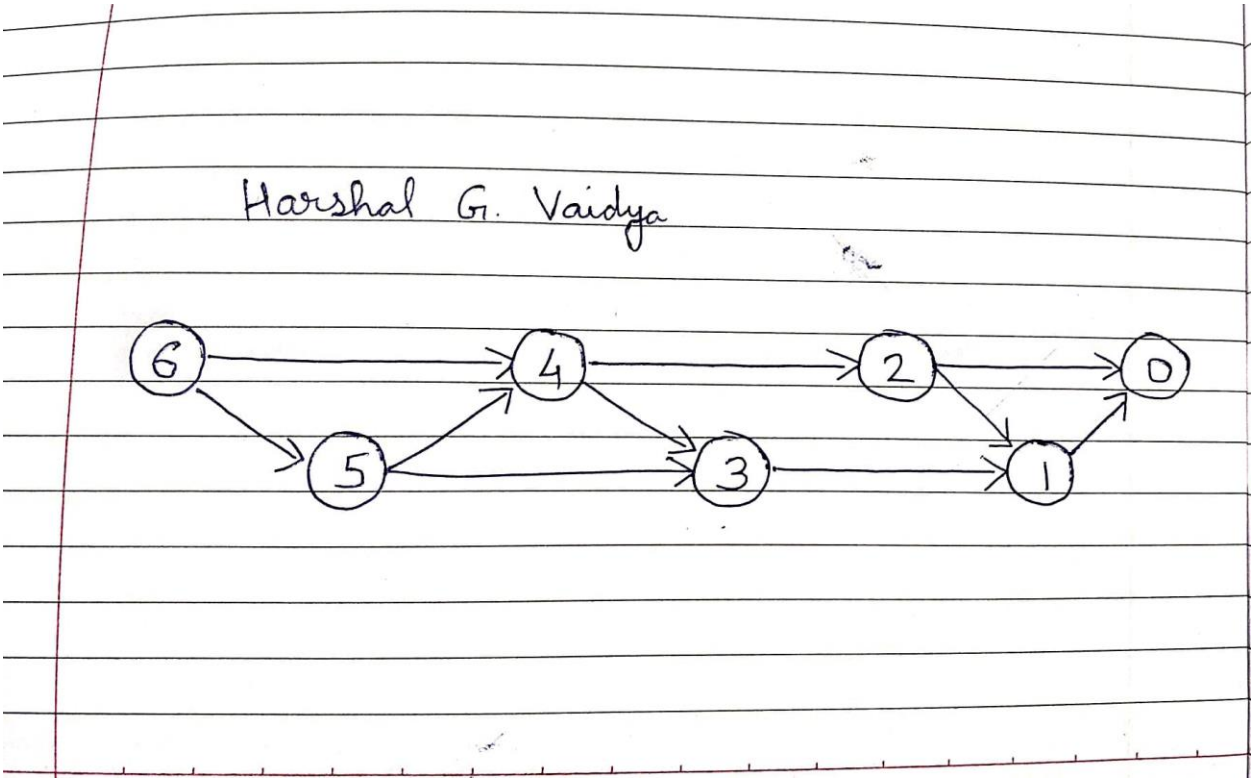
Let's us consider the length of the rod (n) to be 6 i.e.,
n = 6
Now consider the prices to be as follows:-
p1 = 0,
p2 = 1,
p3 = 5,
p4 = 8,
p5 = 2
Now, according to the greedy algorithm, it will consider the height first i.e., p4.
The ratio is pi/i = 8/6.
The left height is 6 - 4 = 2
Therefore, for the heigh of left which is equal to 2, Greedy algorithm will consider the value of p2.
Therefore, the total profit will be 8 + 1 = 9
But a better solution is also possible when we consider as follows:-
p3 + p3 = 6
By doing that, we can get a profit of 10 i.e., 5 + 5 = 10

**Exercise 15.1-5:**

**The Fibonacci numbers are defined by recurrence (3.22). Give an O(n) time dynamic-programming algorithm to compute the n-th Fibonacci number. Draw the subproblem graph. How many vertices and edges are in the graph?**

```
FIBONACCI(n)
  let fib[0. . n] be a new array
  fib[0] = 1
```

```
fib[1] = 1
for i = 2 to n
   fib[i] = fib[i - 1] + fib[i - 2]
return fib[n]
```




The above graph is for the value of n = 6.

Vertex: There are n + 1 vertices in the subproblem graph, i.e., 0, 1, 2, 3, 4, 5, 6

Edges: There are 2n – 2 edges in the subproblem graph.

**Exercise 15.4-1**

**Determine an LCS of ⟨1,0,0,1,0,1,0,1⟩ and ⟨0,1,0,1,1,0,1,1,0⟩**

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 0 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 0 | 1 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 |
| 0 | 1 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 |
| 0 | 1 | 2 | 3 | 4 | 4 | 4 | 5 | 5 | 5 |
| 0 | 1 | 2 | 3 | 4 | 4 | 5 | 5 | 5 | 6 |
| 0 | 1 | 2 | 3 | 4 | 5 | 5 | 6 | 6 | 6 |

The LCS is ⟨1,0,0,1,1,0⟩ or ⟨1,0,1,0,1,0⟩.