Algorithms - Assignment-2 (Dynamic Programming)
Due by 30 September

September 23, 2019

- Try to solve the three problems given using dynamic programming(DP) strategy.
- Understand the recursion involved in your solution and express it as a recurrence relation; remember to set the base conditions.
- **Trace your solution on a small example (building the table/memoization)**
- Submit a hand-written document clearly indicating your roll number.

---

1. There are n bike-rental stops along your office route. At any of the stops you can rent a bike and return the same at any of the next(forward) stops. (It is one-way traffic only and you cannot go backward). For each possible departure point i and each possible arrival point j the cost of the rent from i to j is known. Note that, it is possible that the cost of renting for a direct route from i to j is higher than the total cost of a series of shorter rentals. So it may be advantageous to hop on one bike, make a short trip, return and take a next bike and so on. There is no extra charge for changing the bikes in this way.

   Give an efficient algorithm to determine the minimum cost of a trip by the bike from each possible departure point i to each arrival point j. Also write the sequence of rentals that gives the optimal solution.

2. Given two text strings A of length n and B of length m, to transform A into B using a minimum number of operations of the following types: delete a character from A, insert a character into A, or change some character in A into a new character. The minimal number of such operations required to transform A into B is called the edit distance between A and B. For example, the edit distance(abbc, xbbacd) = 3 (change a to x; insert a after bb; insert d after c)

   Find a recurrence relation that is useful to find the edit distance of two strings A and B. Take an example of two strings of which one is your own name, and using this relation, fill the table for the minimum number of operations.

3. Suppose there are n jobs and k workers. Assume that all the workers are equally efficient. The time taken for each of the jobs is given as $t_1, t_2, \ldots t_n$. The n jobs are to be assigned to the k workers such that the maximum total time taken by a worker is minimized. Provide an algorithm for $k = 2$.

   For example, if $k = 2$ and $n = 6$ and $T = (10, 20, 30, 40, 50, 60)$ then the solution $(10, 20, 30, 40)$ and $(50, 60)$ has maximum total time taken being 110.