Indian Institute of Technology, Madras
Department of Applied Mechanics & Biomedical Engineering
Computational Tools: Algorithms, Data Structures and Programs - ID6105

Assignment set - III

To be submitted by: Monday 29$^{\text{th}}$ September, 2025

1 Problem 1

A n by n Toeplitz matrix is a n by n matrix T such that $T[i,j] = T[i-1,j-1] = t_{i-j}, \forall\ 2 \leq i,j \leq n$. Thus the matrix is completely determined by $2n-1$ values given by $t_k, -n+1 \leq k \leq n-1$, which are along the first row and first column. Consider the following questions,

(a) Consider the multiplication of a Toeplitz matrix T of size n by n with a vector of size n by 1. Find the number of operations that need to be performed.

(b) Generalize the divide and conquer multiplication algorithm for the Toeplitz matrix times a column vector. What will be the resulting complexity?

(c) Implement this in python and show the complexity numerically.

---

2 Problem 2

Shortest path on a grid: Suppose you are setting up a robot to do rescue operations on a horizontal floor. The robot can only either move right or down on a grid and each grid has an associated cost for operation. Consider the following questions,

(a) Consider a horizontal grid 3×3 given below,

1	3	1
1	5	1
4	2	1

with the numbers indicating the cost of operations passing through the particular element on the grid. Find the optimal path from the top left to the bottom right of the grid.

(b) For a general n by n grid, find the number of operations required to get the optimal path using brute force search.

(c) For the above example use a dynamic programming based approach to write a recurrence relation, and then compute the DP table.

(d) Consider a more complex table given below, find the DP table and the optimal path from the top left to the bottom right.

1	4	8	5	7	10	3	7
8	5	4	8	8	3	6	5
2	8	6	2	5	1	10	6
9	1	10	3	7	4	9	3
5	3	7	5	9	7	2	4
9	2	10	9	10	5	2	4
7	8	3	1	4	2	8	4
2	6	6	10	4	6	2	1

.

(e) For the above optimal path solution, show that the subpaths are also optimal. Thus illustrating the Bellman principle of optimality.

(f) Write a python code to generate a general n by n table of random numbers and code the DP table algorithm to find the optimal solution. Find the complexity scaling.

---

3 Problem 3

Coin change problem: Suppose you are setting up an automatic machine to replace the cashier desks at supermarkets. In order to minimise the queue waiting times, you are tasked with finding the minimal set of coins to return as change for any given amount. Given the denominations as $[1, 2, 5, 10, 20]$, answer the following questions,

(a) Given an amount, let's say 38, find the optimal set of change coins that minimises the total number of coins using brute force method.

(b) Use a greedy algorithm, for which at each step we choose the coins which reduces the total number by the maximum.

(c) Write a python code to implement this algorithm, for any arbitrary amount (which is a positive integer).

(d) Does the greedy algorithm always give the optimal solution? How about a set of coins with denomination $[1, 3, 4]$?

---

4 Problem 4

Area estimation of irregular shapes: Suppose you are asked to estimate the area of a land parcel for an upcoming residential building. The plot is irregular in shape due to existing buildings/structures. We will use a randomised algorithm to find the area of the irregular plot.

(a) Let's first take an example of a polygon. Construct a rectangular shape that encloses the polygon and randomly drop points into the rectangle. The area of the polygon is then estimated to be,

$$A_{\text{polygon}} \approx A_{\text{rectangle}} \frac{M}{N}, \tag{1}$$

where M is the total number of points that fell into the polygon and N is the total number of points inside the rectangle.

(b) Write a python code and plot the error as a function of N. Explain the behaviour.

(c) Now consider the arbitrary shape 1, repeat the same exercise.

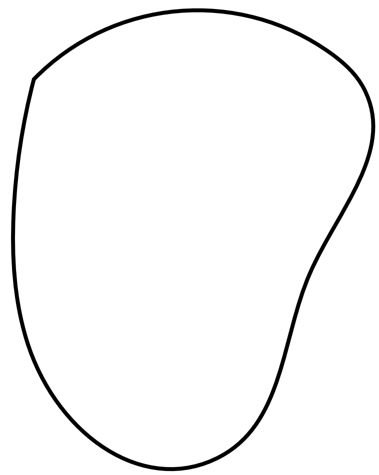

Figure 1: Random shape.

---

5 Problem 5

Trajectory optimisation: Suppose you are hired as an aerospace engineer to oversee the transfer of orbit of a satellite. To transfer the orbit of the satellite, we fire one of the engines and that costs us fuel, thus we want to optimise the amount of fuel spent. We define the cost as,

$$J(u) = \sum_{i=1}^{N-1} u_k^2 \tag{2}$$

where $u_0, u_1, u_2, \dots u_{N-1}$ is the set of control inputs. $J(u)$ is the total cost of the operation. Let us answer the following,

(a) Consider the case of dynamics given by $x_{k+1} = x_k + dt \times u_k$ with dt being fixed. Starting from $x_0 = 0$, we want to reach $x_N = 1.0$ in N number of steps. What is the optimal strategy that minimises $J(u)$? What is the minimum $J(u)$? Take $dt \ll 1, N \gg 1$.

(b) We take the full dynamics now given by both the position and velocity fields,

$$x_{k+1} = x_k + dt \times u_k, \qquad u_{k+1} = u_k + dt \times a_k. \tag{3}$$

The initial state is given by $x_0 = 0, u_0 = 0$, the final state is given by $x_N = 1, u_N = 0$. We define the cost function as,

$$J = (x_N - 1.0)^2 + u_N^2 + \sum_{k=0}^{N-1} a_k^2 \tag{4}$$

Find the optimal set of control inputs $a_0, a_1, a_2, \dots, a_{N-1}$ which minimises the cost function.

(c) Write a python program to do the above tasks and add the following constraint that $a_k \leq a_{\max}$.

(d) Suppose there are space debris that need to be avoided at positions $x = 0.4,\ 0.6$, redefine the cost function to add a penalty and redo the optimisation.