

PSO #5 Week 7

Warm-up

Given n positive integers stored in array **A** and a value **T**, determine if there is a subset **S** whose elements sum to **T**

Warm-up

Let \mathbf{L} be an array of size n containing integers in arbitrary order.

In the Longest Increasing Subsequence problem, the goal is to find the length of the longest subsequence in \mathbf{L} such that all elements of the subsequence are sorted in increasing order.

Once the longest length is known, generate/recover the subsequence.

Question 1

Given an $n \times n$ matrix **A**, find the length of the longest path of adjacent numbers through the matrix. The path can be formed of a sequence of numbers going up, down, left, or right. Adjacent numbers means each number in the sequence of numbers must be either 1 smaller or 1 greater than the two numbers on either side of it that are still part of the path.

Example:

{ 10	13	14	21	23 }
{ 11	9	22	2	3 }
{ 12	8	1	5	4 }
{ 15	24	7	6	20 }
{ 16	17	18	19	25 }

{ 10	13—14	21	23 }
{ 11	9	22	2—3 }
{ 12	8	1	5—4 }
{ 15	24	7—6	20 }
{ 16—17—18—19	25 }		

Question 2

Given a string **S** of length **n**, the longest palindrome is the longest subset of characters such that if one were to extract them from the string in order, then the resulting substring would form a palindrome.

Design an algorithm to determine the length of the longest palindrome

Question 3

Given a rod cut into n pieces, the cost of joining two pieces is twice the length of the left piece plus the length of the right piece. Only adjacent pieces may be joined.

Design an algorithm to join the rod with minimum cost.