Problem Solving 5.
Tuesday, December 7, 2021

Today's Content

1) Largest Subarray Sum
2) Maximum Difference
3) Continuous Array
4) Longest Bitonic Subsequence
5) Longest Increasing Subsequence
6) Longest Common Subsequence
7) Longest Palindromic Subsequence
8) Longest Palindromic Substring

1) Largest Subarray Sum

Given an array $A[\,]$ of size N, find the maximum sum of a contiguous subarray.

$A[\,] = \{-3, 5, 4, 2, -3, 7, 1, 2\}$

Ans: $\{5, 4, 2, -3, 7, 1, 2\}$

Brute force approach

$A[\,] = \{-3, 5, 4, 2, -3, 7, 1, 2\}$

$\{-3\}, \{-3, 5\}, \{-3, 5, 4\}, \{-3, 5, 4, 2\}, \{-3, 5, 4, 2, -3\}, \{-3, 5, 4, 2, -3, 7\}, \{-3, 5, 4, 2, -3, 7, 1\}, \{-3, 5, 4, 2, -3, 7, 1, 2\}$

$\{5\}, \{5, 4\}, \{5, 4, 2\}, \{5, 4, 2, -3\}, \{5, 4, 2, -3, 7\}, \{5, 4, 2, -3, 7, 1\}, \{5, 4, 2, -3, 7, 1, 2\}$

$\{4\}, \{4, 2\}, \{4, 2, -3\}, \{4, 2, -3, 7\}, \{4, 2, -3, 7, 1\}, \{4, 2, -3, 7, 1, 2\}$

$\{2\}, \{2, -3\}, \{2, -3, 7\}, \{2, -3, 7, 1\}, \{2, -3, 7, 1, 2\}$

$\{-3\}, \{-3, 7\}, \{-3, 7, 1\}, \{-3, 7, 1, 2\}$

$\{7\}, \{7, 1\}, \{7, 1, 2\}$

$\{1\}, \{1, 2\}$

$\{2\}$

Time complexity $= O(N^2)$

Kadane's Algorithm

Step 1: Initialize $sum = 0$, $ans = -\infty$

Step 2: Traverse through the array and add each element to sum

Step 3: If $sum > ans$, then update $ans = sum$

Step 4: If $sum < 0$, then reset $sum = 0$

Step 5: Return ans

$A[\,] = \{-3, 5, 4, 2, -3, 7, 1, 2\}$

$sum = 0$, $ans = -\infty$

$sum = -3$, $ans = -3$

$sum = 5$, $ans = 5$

$sum = 9$, $ans = 9$

$sum = 11$, $ans = 11$

$sum = 8$, $ans = 11$

$sum = 15$, $ans = 15$

$sum = 16$, $ans = 16$

$sum = 18$, $ans = 18$

Ans: 18

Time complexity $= O(N)$

2) Maximum Difference

Given $A[\,]$ find the term b & a common $\& \le b$ Element

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, \ldots, a_N$

$a_1 \le a_2$, $a_1 \le a_3$, $a_1 \le a_4$, $a_1 \le a_5$, $a_1 \le a_6$, $a_1 \le a_7$, $a_1 \le a_8$, $\ldots$, $a_1 \le a_N$

$a_2 \le a_3$, $a_2 \le a_4$, $a_2 \le a_5$, $a_2 \le a_6$, $a_2 \le a_7$, $a_2 \le a_8$, $\ldots$, $a_2 \le a_N$

$a_3 \le a_4$, $a_3 \le a_5$, $a_3 \le a_6$, $a_3 \le a_7$, $a_3 \le a_8$, $\ldots$, $a_3 \le a_N$

$a_4 \le a_5$, $a_4 \le a_6$, $a_4 \le a_7$, $a_4 \le a_8$, $\ldots$, $a_4 \le a_N$

$a_5 \le a_6$, $a_5 \le a_7$, $a_5 \le a_8$, $\ldots$, $a_5 \le a_N$

$a_6 \le a_7$, $a_6 \le a_8$, $\ldots$, $a_6 \le a_N$

$a_7 \le a_8$, $\ldots$, $a_7 \le a_N$

$\ldots$

$a_N \le a_N$

$A[\,] = \{-3, 5, 4, 2, -3, 7, 1, 2\}$

$\{-3, 5\}, \{-3, 4\}, \{-3, 2\}, \{-3, -3\}, \{-3, 7\}, \{-3, 1\}, \{-3, 2\}$

$\{5, 4\}, \{5, 2\}, \{5, -3\}, \{5, 7\}, \{5, 1\}, \{5, 2\}$

$\{4, 2\}, \{4, -3\}, \{4, 7\}, \{4, 1\}, \{4, 2\}$

$\{2, -3\}, \{2, 7\}, \{2, 1\}, \{2, 2\}$

$\{-3, 7\}, \{-3, 1\}, \{-3, 2\}$

$\{7, 1\}, \{7, 2\}$

$\{1, 2\}$

Time complexity $= O(N^2)$

Optimized approach

Step 1: Create an array $B[\,]$ of size N

Step 2: Traverse through the array $A[\,]$ and add the elements to $B[\,]$ in sorted order

Step 3: Traverse through the array $A[\,]$ and find the difference between each element and the element present in $B[\,]$ using binary search

Step 4: Return the maximum difference

$A[\,] = \{-3, 5, 4, 2, -3, 7, 1, 2\}$

$B[\,] = \{-3, -3, 1, 2, 2, 4, 5, 7\}$

$5 - -3 = 8$

$4 - -3 = 7$

$2 - -3 = 5$

$-3 - -3 = 0$

$7 - -3 = 10$

$1 - -3 = 4$

$2 - -3 = 5$

Ans: 10

Time complexity $= O(N \log N)$

3) Continuous Array

Given an array $A[\,]$ of size N consisting of 0's and 1's, find the maximum length of a contiguous subarray with equal number of 0's and 1's.

$A[\,] = \{0, 1, 0, 1, 0, 1, 1\}$

$\{0, 1\}, \{0, 1, 0, 1\}, \{0, 1, 0, 1, 0, 1\}, \{0, 1, 0, 1, 0, 1, 1\}$

$\{1, 0\}, \{1, 0, 1, 0, 1\}, \{1, 0, 1, 0, 1, 1\}$

$\{0, 1, 0, 1\}, \{0, 1, 0, 1, 1\}$

$\{1, 0, 1\}, \{1, 0, 1, 1\}$

$\{0, 1\}, \{1, 1\}$

$\{1\}$

Time complexity $= O(N^2)$

Optimized approach

Step 1: Initialize $sum = 0$, $ans = 0$

Step 2: Traverse through the array and add each element to sum

Step 3: If $sum = 0$, then update $ans = i + 1$

Step 4: Return ans

$A[\,] = \{0, 1, 0, 1, 0, 1, 1\}$

$sum = 0$, $ans = 0$

$sum = -1$, $ans = 0$

$sum = 0$, $ans = 2$

$sum = 1$, $ans = 2$

$sum = 0$, $ans = 4$

$sum = 1$, $ans = 4$

$sum = 2$, $ans = 4$

Ans: 4

Time complexity $= O(N)$

4) Longest Bitonic Subsequence

Given an array $A[\,]$ of size N, find the longest subsequence such that elements in the subsequence are first sorted in increasing order and then sorted in decreasing order.

$A[\,] = \{1, 2, 5, 3, 2\}$

$\{1, 2, 5\}, \{1, 2, 3\}, \{1, 2, 2\}, \{1, 5, 3\}, \{1, 5, 2\}, \{1, 3, 2\}, \{2, 5, 3\}, \{2, 5, 2\}, \{2, 3, 2\}, \{5, 3, 2\}$

Ans: $\{1, 2, 5, 3, 2\}$

Brute force approach

Step 1: Create an array $B[\,]$ of size N

Step 2: Traverse through the array $A[\,]$ and find the longest increasing subsequence ending at i and store it in $B[i]$

Step 3: Create an array $C[\,]$ of size N

Step 4: Traverse through the array $A[\,]$ and find the longest decreasing subsequence starting at i and store it in $C[i]$

Step 5: Traverse through the array $A[\,]$ and find the maximum of $B[i] + C[i] - 1$

$A[\,] = \{1, 2, 5, 3, 2\}$

$B[\,] = \{1, 2, 3, 2, 2\}$

$C[\,] = \{3, 2, 2, 2, 1\}$

$1 + 3 - 1 = 3$

$2 + 2 - 1 = 3$

$3 + 2 - 1 = 4$

$2 + 2 - 1 = 3$

$2 + 1 - 1 = 2$

Ans: 4

Time complexity $= O(N^2)$

5) Longest Increasing Subsequence

Given an array $A[\,]$ of size N, find the length of the longest increasing subsequence.

$A[\,] = \{10, 22, 9, 33, 21, 50, 41, 60, 80\}$

$\{10, 22, 33, 50, 60, 80\}$

$\{10, 22, 41, 60, 80\}$

$\{10, 22, 41, 60\}$

$\{10, 22, 41, 80\}$

$\{10, 22, 50, 60, 80\}$

$\{10, 22, 50, 80\}$

$\{10, 22, 60, 80\}$

$\{10, 22, 60\}$

$\{10, 22, 80\}$

$\{10, 22\}$

$\{10, 9\}$

$\{10, 33\}$

$\{10, 21\}$

$\{10, 50\}$

$\{10, 41\}$

$\{10, 60\}$

$\{10, 80\}$

$\{10\}$

Ans: 5

Brute force approach

Step 1: Create an array $B[\,]$ of size N

Step 2: Traverse through the array $A[\,]$ and find the longest increasing subsequence ending at i and store it in $B[i]$

Step 3: Return the maximum value in $B[\,]$

$A[\,] = \{10, 22, 9, 33, 21, 50, 41, 60, 80\}$

$B[\,] = \{1, 2, 1, 3, 2, 4, 3, 5, 6\}$

Ans: 6

Time complexity $= O(N^2)$

6) Longest Common Subsequence

Given two strings A & B, find the length of the longest common subsequence.

$A = \text{"abcde"}$, $B = \text{"ace"}$

$\{a, c, e\}$

Ans: 3

Brute force approach

Step 1: Create all possible subsequences of A

Step 2: Create all possible subsequences of B

Step 3: Find the longest common subsequence

Step 4: Return the length of the longest common subsequence

$A = \text{"abcde"}$, $B = \text{"ace"}$

$\{a\}, \{a, b\}, \{a, b, c\}, \{a, b, c, d\}, \{a, b, c, d, e\}$

$\{a, c\}, \{a, b, c\}, \{a, b, c, d\}, \{a, b, c, d, e\}$

$\{a, c, e\}, \{a, b, c, d\}, \{a, b, c, d, e\}$

$\{a, c, e\}, \{a, b, c, d, e\}$

$\{a, c, e\}$

Time complexity $= O(2^N)$

Optimized approach

Step 1: Create an array $B[\,]$ of size N

Step 2: Traverse through the array $A[\,]$ and find the longest increasing subsequence ending at i and store it in $B[i]$

Step 3: Create an array $C[\,]$ of size N

Step 4: Traverse through the array $A[\,]$ and find the longest decreasing subsequence starting at i and store it in $C[i]$

Step 5: Traverse through the array $A[\,]$ and find the maximum of $B[i] + C[i] - 1$

$A[\,] = \{1, 2, 5, 3, 2\}$

$B[\,] = \{1, 2, 3, 2, 2\}$

$C[\,] = \{3, 2, 2, 2, 1\}$

$1 + 3 - 1 = 3$

$2 + 2 - 1 = 3$

$3 + 2 - 1 = 4$

$2 + 2 - 1 = 3$

$2 + 1 - 1 = 2$

Ans: 4

Time complexity $= O(N^2)$

7) Longest Palindromic Subsequence

Given a string A, find the length of the longest palindromic subsequence.

$A = \text{"bbbab"}$

$\{b, b, b, b\}$

Ans: 4

Brute force approach

Step 1: Create all possible subsequences of A

Step 2: Find the longest palindromic subsequence

Step 3: Return the length of the longest palindromic subsequence

$A = \text{"bbbab"}$

$\{b\}, \{b, b\}, \{b, b, b\}, \{b, b, b, a\}, \{b, b, b, a, b\}$

$\{b, b\}, \{b, b, b\}, \{b, b, b, a\}, \{b, b, b, a, b\}$

$\{b, b, b\}, \{b, b, b, a\}, \{b, b, b, a, b\}$

$\{b, b, b, a\}, \{b, b, b, a, b\}$

$\{b, b, b, a, b\}$

Time complexity $= O(2^N)$

Optimized approach

Step 1: Create an array $B[\,]$ of size N

Step 2: Traverse through the array $A[\,]$ and find the longest palindromic subsequence ending at i and store it in $B[i]$

Step 3: Create an array $C[\,]$ of size N

Step 4: Traverse through the array $A[\,]$ and find the longest palindromic subsequence starting at i and store it in $C[i]$

Step 5: Traverse through the array $A[\,]$ and find the maximum of $B[i] + C[i] - 1$

$A = \text{"bbbab"}$

$B[\,] = \{1, 2, 3, 1, 1\}$

$C[\,] = \{4, 3, 2, 1, 1\}$

$1 + 4 - 1 = 4$

$2 + 3 - 1 = 4$

$3 + 2 - 1 = 4$

$1 + 1 - 1 = 1$

$1 + 1 - 1 = 1$

Ans: 4

Time complexity $= O(N^2)$

8) Longest Palindromic Substring

Given a string A, find the longest palindromic substring.

$A = \text{"babad"}$

$\text{"bab"}, \text{"aba"}$

Brute force approach

Step 1: Create all possible substrings of A

Step 2: Find the longest palindromic substring

Step 3: Return the length of the longest palindromic substring

$A = \text{"babad"}$

$\text{"b"}, \text{"ba"}, \text{"bab"}, \text{"baba"}, \text{"babad"}$

$\text{"a"}, \text{"ab"}, \text{"abad"}$

$\text{"b"}, \text{"ba"}, \text{"bad"}$

$\text{"a"}, \text{"ad"}$

"d"

Time complexity $= O(N^3)$

Optimized approach

Step 1: Create an array $B[\,]$ of size N

Step 2: Traverse through the array $A[\,]$ and find the longest palindromic substring ending at i and store it in $B[i]$

Step 3: Create an array $C[\,]$ of size N

Step 4: Traverse through the array $A[\,]$ and find the longest palindromic substring starting at i and store it in $C[i]$

Step 5: Traverse through the array $A[\,]$ and find the maximum of $B[i] + C[i] - 1$

$A = \text{"babad"}$

$B[\,] = \{1, 3, 1, 1, 1\}$

$C[\,] = \{3, 2, 1, 1, 1\}$

$1 + 3 - 1 = 3$

$3 + 2 - 1 = 4$

$1 + 1 - 1 = 1$

$1 + 1 - 1 = 1$

$1 + 1 - 1 = 1$

Ans: 4

Time complexity $= O(N^2)$