

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Design and Analysis of Algorithms**

**TERM: March - July 2022**

**PROJECT SYNOPSIS**

**PROJECT TEAM MEMBERS**

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **USN** | **Name** |
| 1. | 1MS20CS008 | ADITYA SHARMA |
| 2. | 1MS20CS011 | AMAN BHATNAGAR |
| 3. | 1MS20CS037 | DANISH MAHAJAN |
| 4. | 1MS20CS044 | DHRUV KHANDELWAL |

M.S. RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)

**Title of the Project:**

Business analysis using kadane’s algorithm.

**Problem Statement:**

Business analysis where you need to find out the growth of a company in a particular duration of time by taking the price of stocks into consideration.

**Objectives:**

* To find the period of maximum growth of a company, given their monthly stock price.
* Using algorithm which have time complexity of O(n).
* Using algorithm which have space complexity of O(1).

**Literature Review:**

**[1]. In this research we find that this problem is solved by most simple method i.e brute force. In this approach, 3 for loops are used to solve this problem. The most outer loop is used to choose the starting point of subarray, the middle loop is used to choose the ending point of subarray and the most inner loop is used to find the sum of subarray.**

**[2].** **In this research we find that the solution is solved in better way using 2 for loops. This approach is better than previous approach in which we use 3 for loops. This approach is also come under brute force. In this outer loop is used to choose the starting point of subarray and inner loop is used to find the sum of subarray from starting point to end point and takes the maximum sum in this loop and compare it with previous maximum sum of subarray**

**[3].** **In this research we find that this problem can solved in a better way than brute force. This approach, is divide and conquer. In this approach we divide the array into two equal parts and recursively find the maximum subarray sum of the left and right part and also the maximum sum subarray between the left and right arrays.**

**[4].** **In this research we find that this problem is solved by using Dynamic programming technique. In this approach, 2 for loop are used, a loop is used to find the max sum at ith elements and store it in a different array(say Max) of size n and second loop(not inner loop ,different loop) is used to find the biggest sum in the Max array**

**[5].** **In this research we find that this problem is solved in most optimal solution (Kadane’s Algorithm). In this approach we use two variables lets, say X,Y. now we use for loop, inside it we find the sum of all elements upto ith elements and compare it with Y. if Y is less than X the Y become X, otherwise X become zero and when the for loop end return the value of Y.**

**Research Gaps Identified:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.NO | ALGO. | TIME COM. | SPACE COMP. | DISADVANTAGES |
| [1] | Brute force | O(N^3) | O(1) | A better time complexity can be found like O(n^2). |
| [2] | Brute force | O(N^2) | O(1) | A better time complexity can be found like O(n). |
| [3] | Divide and conquer | O(NlogN) | O(logN) | A better time complexity can be found O(n). |
| [4] | Dynamic program. | O(N) | O(N) | A Better space complexity can be found O(1). |
| [5] | Kadane’s algorithm | O(N) | O(1) | If all elements in an array are negative it gives answer zero, instead of some negative number. |

**Proposed System:**

Maxso= max sum of stocks price in given data.

End

Return maxso

Take net profit in an array from user

N<=1

**yes**

**No**

**Run n-1 times**

Maxso is less maxend

**Yes No**

**End of for loop**

**Algorithm Used: Kadane’s Algorithm**

***pseudocode***

**Initialize:**

max\_so\_far = INT\_MIN

max\_ending\_here = 0

Loop for each element of the array

(a) max\_ending\_here = max\_ending\_here + a[i]

(b) if(max\_so\_far < max\_ending\_here)

max\_so\_far = max\_ending\_here

(c) if(max\_ending\_here < 0)

max\_ending\_here = 0

return max\_so\_far

**A better approach to kadane's algorithm:**

**In simple kadane's algorithm if all elements all negative it gives the answer 0 instead of least negative number**. Kadane's algorithm can be improved, in such a way that if we got the answer 0 then we can do this:

if (max\_so\_far == 0)

max\_so\_far = maxelement;

**Data Structure Used**: **ARRAY**

* The complexity for accessing, pushing or popping a value in an array is **O(1).**
* Filled arrays take up less memory than linked lists.
* In array, each element is independent and can be accessed using, it's index value.

**Analysis of the chosen Algorithm**

* each element has been visited only once.
* Time Complexity = O(n)
* Space Complexity = O(1)

**What contribution to the society, would the project make?**

* In business analysis where you need to find out the duration of time where the company experienced the maximum growth.
* Count maximum number of cars parked at the same time.
* used in genomic sequence analysis to identify important segments of protein sequences like GC-rich regions, and regions of high charge.
* In computer vision, they find their use in detecting the brightest area in bitmap images.

**References:**

[1]. [**https://www.codingninjas.com/blog/2020/09/17/a-quick-look-at- kadanes-algorithm/**](https://www.codingninjas.com/blog/2020/09/17/a-quick-look-at-%20%20kadanes-algorithm/)

**~Neelakshi Lahiri**

[2]. [**https://medium.com/@rsinghal757/kadanes-algorithm-dynamic-programming-how-and-why-does-it-work-3fd8849ed73d**](https://medium.com/@rsinghal757/kadanes-algorithm-dynamic-programming-how-and-why-does-it-work-3fd8849ed73d)

**~ Ruhal Singhal**

[3]. [**https://afteracademy.com/blog/maximum-subarray-sum**](https://afteracademy.com/blog/maximum-subarray-sum)

~AfterAcademy

[4]. [**https://www.enjoyalgorithms.com/blog/maximum-subarray-sum**](https://www.enjoyalgorithms.com/blog/maximum-subarray-sum) **~ Shubham Gautam**

**[5].** [**http://theoryofprogramming.com/2016/10/21/dynamic-programming-kadanes-algorithm/**](http://theoryofprogramming.com/2016/10/21/dynamic-programming-kadanes-algorithm/)

**~ Vamsi Sangam**