Design And Analysis Of Algorithms-Assignment 4
Group-2

ADITYA RAJ(IIT2017005)
MOHIT PAHUJA(IIT2017007)
MAYANK MRINAL(IIT2017006)
ARASHPREET SINGH (IIT2017008)

***Abstract*— This document is complete design and analysis of algorithm of finding an optimal way if you can reach a given number x from 0 when you move i steps.**

I. INTRODUCTION

In this problem we have to find an optimal way to reach at point x from 0 by moving i distance in i^{th} step. For example 3 takes 2 steps (0, 1) (1, 3) and 4 takes 3 steps (0, -1), (-1, 1) (1, 4). So we are finding both minimum number of moves and the path. For this we have design only one algorithm .

This report further contains::
II. Algorithm Design.
III.Algorithm Analysis
IV.Experimental Study.
V.Conclusions.
VI.References.

II. ALGORITHM DESIGN

A. Algorithm 1

Idea is to move in one direction as long as possible, this will give minimum moves. Starting at 0 first move takes us to 1, second move takes us to 3 (1+2) position, third move takes us to 6 (1+2+3) position, ans so on; So for finding target we keep on adding moves until we find the nth move such that 1+2+3+...+n>=target. Now if sum (1+2+3+...+n) is equal to target the our job is done, i.e we'll need n moves to reach target. Now next case where sum is greater than target. Find the difference by how much we are ahead, i.e sum – target. Let the difference be d = sum – target.
If we take the i-th move backward then the new sum will become (sum – 2i), i.e 1+2+3+...-x+x+1...+n. Now if sum-2i = target then our job is done. Since, sum – target = 2i, i.e difference should be even as we will get an integer i flipping which will give the answer. So following cases arise.

$Case1$: Difference is even then answer is n, (because we will always get a move flipping which will lead to target).

$Case2$: Difference is odd, then we take one more step, i.e add n+1 to sum and now again take the difference. If difference is even the n+1 is the answer else we would have to take one more move and this will certainly make the difference even then answer will be n+2.

Explanation : Since difference is odd. Target is either odd or even. case 1: n is even (1+2+3+...+n) then adding n+1 makes the difference even. case 2: n is odd then adding n+1 doesn't makes difference even so we would have to take one more move, so n+2.

Algorithm 1

```
printv(vector v)
for i=0,i<v.size,i++
v[i]

main()
cin←target
sum←0
i←1
ctr←0
while sum≤ target do
    sum<sum+i
  if sum==target then
      ctr←1
    i++
    else
  end
  if ctr==0 then
    s←0
    i←1
    while sum≤ target do
  s<s+i
  i++
  if s≤ target then
    v.push_back(s)
  else
    c
  end
  out←no. of moves: v.size
  printv(v)
  return 0
    end
  end if
  =0
```

```
    while abs(sum-target)%2!=0 do
        sum ←sum+i
      i++
    end
    rev←(sum-target)>>1
sum ←0
step ←1

    while sum!=target do
        if step!=rev then
            sum<sum+step
        else
        end
        sum←sum- step
      v.push_back(sum)
      step++
    end
    cout←no. of moves: v.size
printv(v)
return 0
```

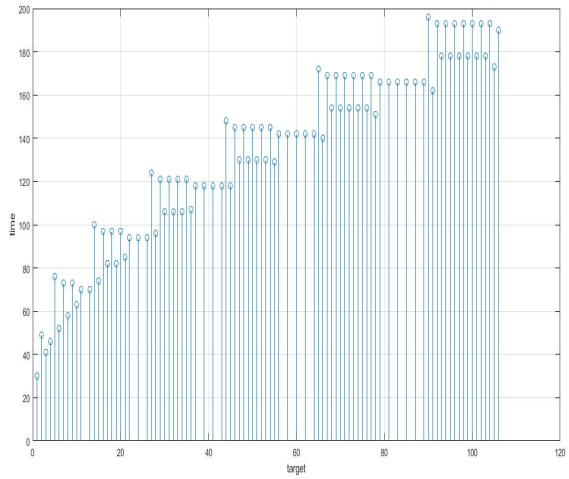
III. ALGORITHM ANALYSIS

A. Time Complexity

- *We have calculated the time consumed in each step in terms of number of computations,as given in the above pseudo code.*
- *Thus for general value of x let say n , the time complexity is going to be in order of $2n$.*
- *The best case , worst case and average case time complexity will be same i.e. $O(n)$.*

B. Space Complexity

- *Since the number of memory location depends on the the variable x since we are storing the path in vector. In worst case the size of the requirement will be $O(x)$.*



Thus the graph depicts, our calculation, that is the algorithm takes polynomial of order n..

IV. EXPERIMENTAL STUDY

The graph depicts that the plot between the target i.e. n and time which comes out to be approximately in the order of $2n$.

The integrated Development environment of C++ is used for processing the algorithm and graphical analysis is done using MATLAB stem function.

V. CONCLUSIONS

In this document we conclude that our algorithm has a polynomial time complexity with order of $2n$ for all the cases i.e. best,worst and average case . The space complexity is also of the order n(where n is the size of target number.) Therefore both time and space complexity is $O(n)$.

VI. REFERENCES

1.https://www.geeksforgeeks.org/find-minimum-moves-reach-target-infinite-line/ 2.[CLR96] Thomas H. Cormen, Charles E. Leiserson, and Ronald L Rivest. Introduction to algorithms. The MIT press, 2nd edition, 1996.
3.[DW96] Nell Dale and Henry M. Walker. Abstract data types: specifications, implementations, and applications. D. C. Heath and Company, Lexington, MA, USA, 1996.