Constructive Comparison Between Two Algorithms

-AbssZy

Table of Content

Abstract

In this project our problem statement was to calculate the minimum amount of profit that can be ensured to an investor in a cashflow. Suppose there are certain investments being made in a share market type of background. The weights in the graphs indicate the profit or the loss after the investment at every node. Positive weights indicate profit and negative weights indicate loss. By using this algorithm, we can find the minimum amount of profit which an investor will have at each node. We have used two algorithms using two different techniques

1. Bellman ford algorithm using dynamic programming.
2. A* algorithm using heuristic search.

We have solved our problem statement using both these algorithms and have made a comparison between the time complexities of the optimal solutions given by both the algorithms.

Introduction

In today's world investing in share market has become a common phenomenon for investing money and making quick money. Though investing in share markets gives quick and good profit returns but there are also high risks for losses. Thus, investors do not feel very convincing to invest in share market. In this project our aim was to solve this issue. Using graphs, we can map the transactions between various lenders into a pictorial description using graphs where the indicate the profit or the loss after the investment and the nodes represent the money lenders. The investor can find out the minimum amount of profit which he/she will definitely get at each node using these techniques.

Time Complexity

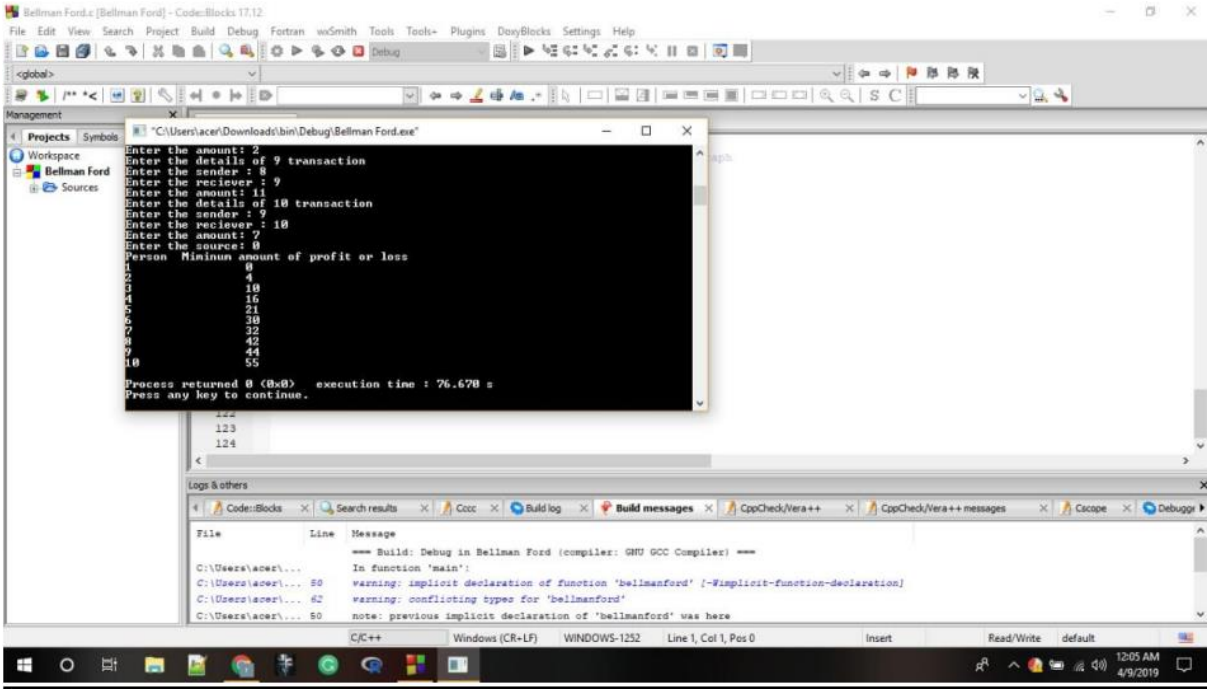
This solution uses two algorithms: -

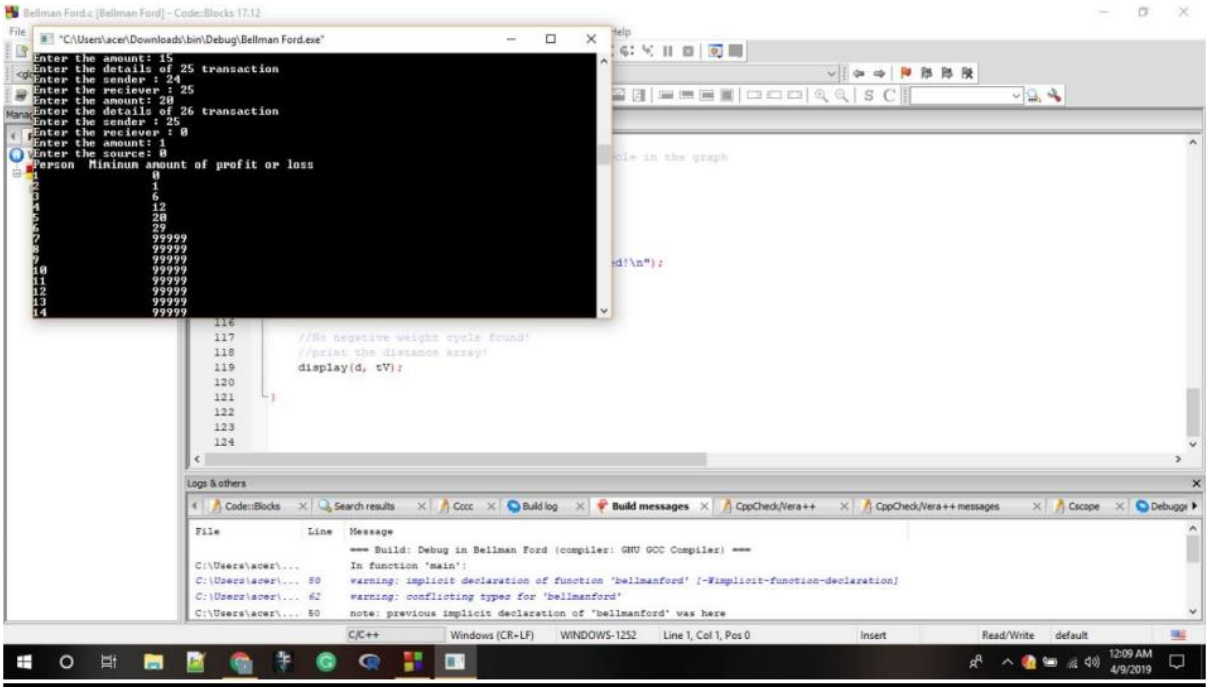
Bellman–Ford algorithm: This algorithm runs in O (|V|. |E|) where |V| and |E| are the number of vertices and edges respectively.

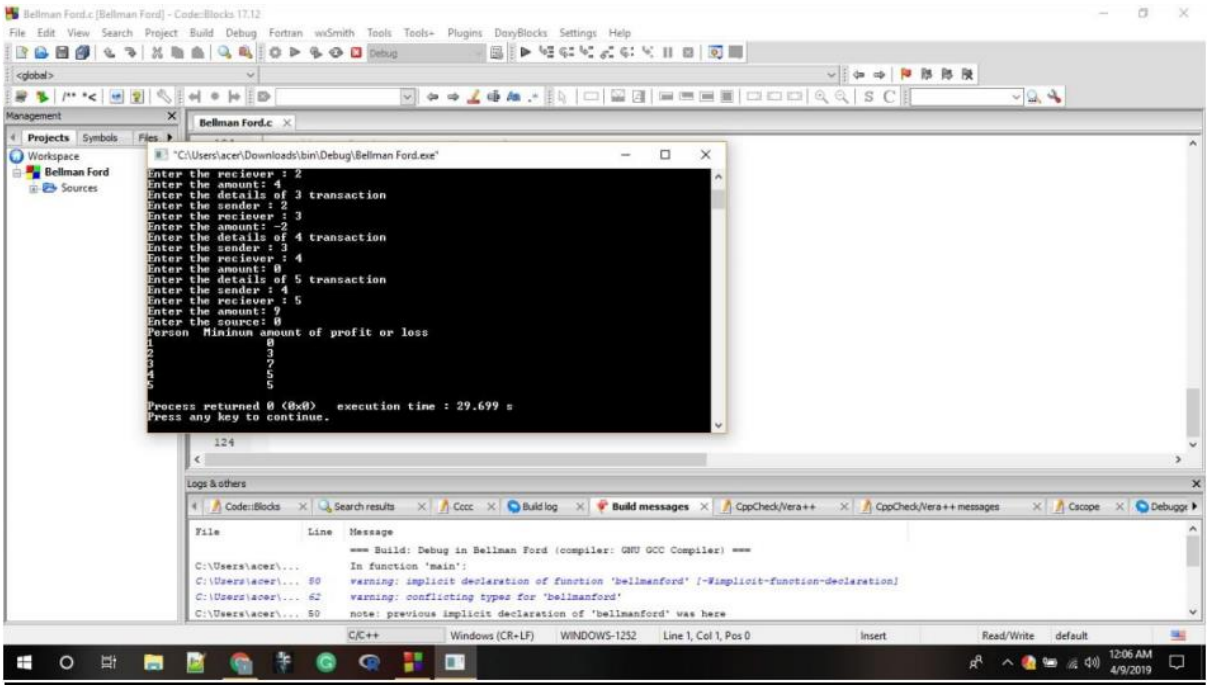
A Star algorithm: The time complexity of A* depends on the heuristic. In the worst case of an unbounded search space, the number of nodes expanded is exponential in the depth of the solution (the shortest path) d: O(b^d), where b is the branching factor (the average number of successors per state). This assumes that a goal state exists at all, and is reachable from the start state; if it is not, and the state space is infinite, the algorithm will not terminate.

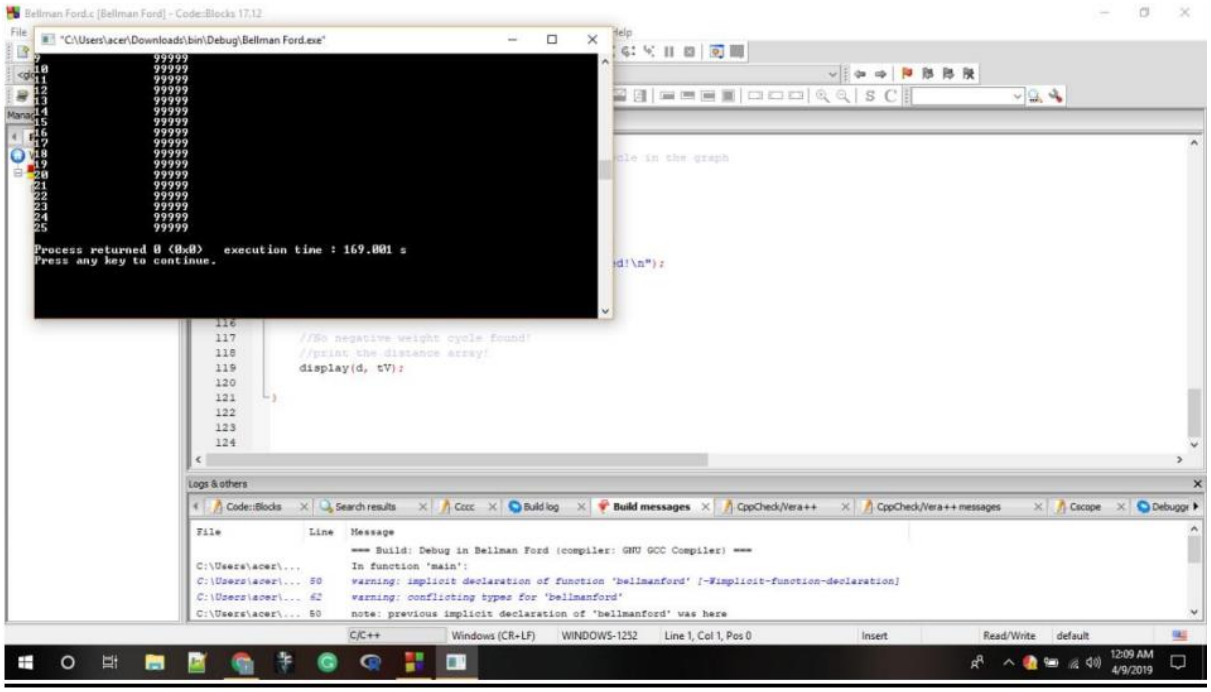
Code Output

1. Bellman Ford Algorithm

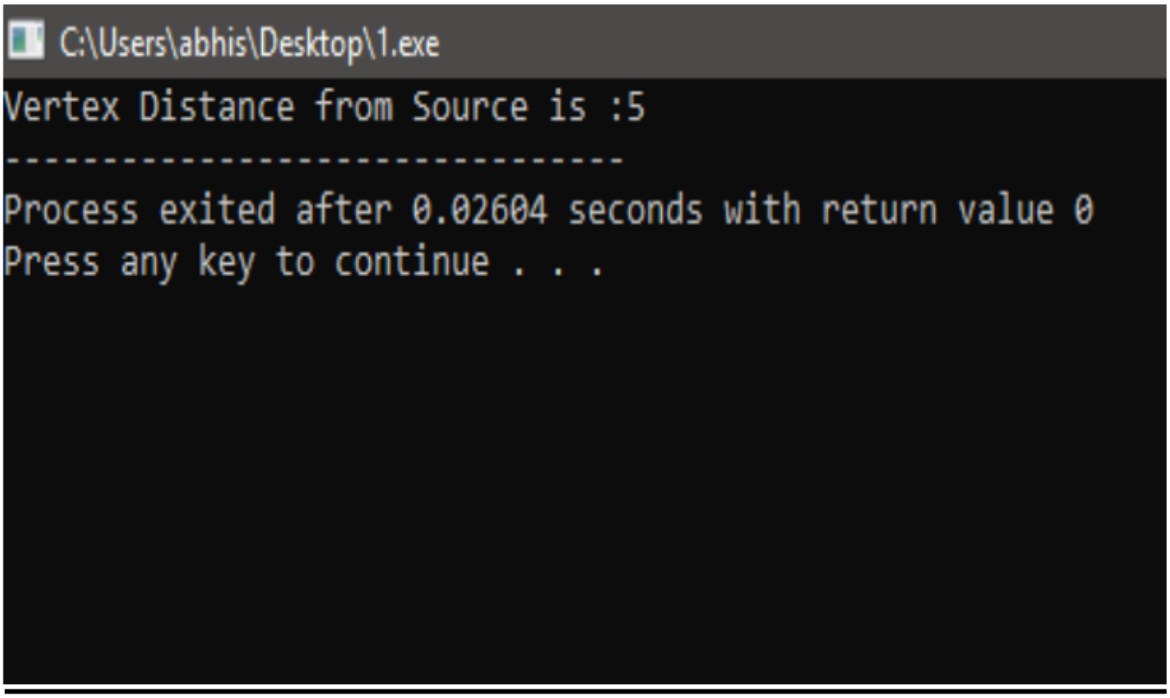




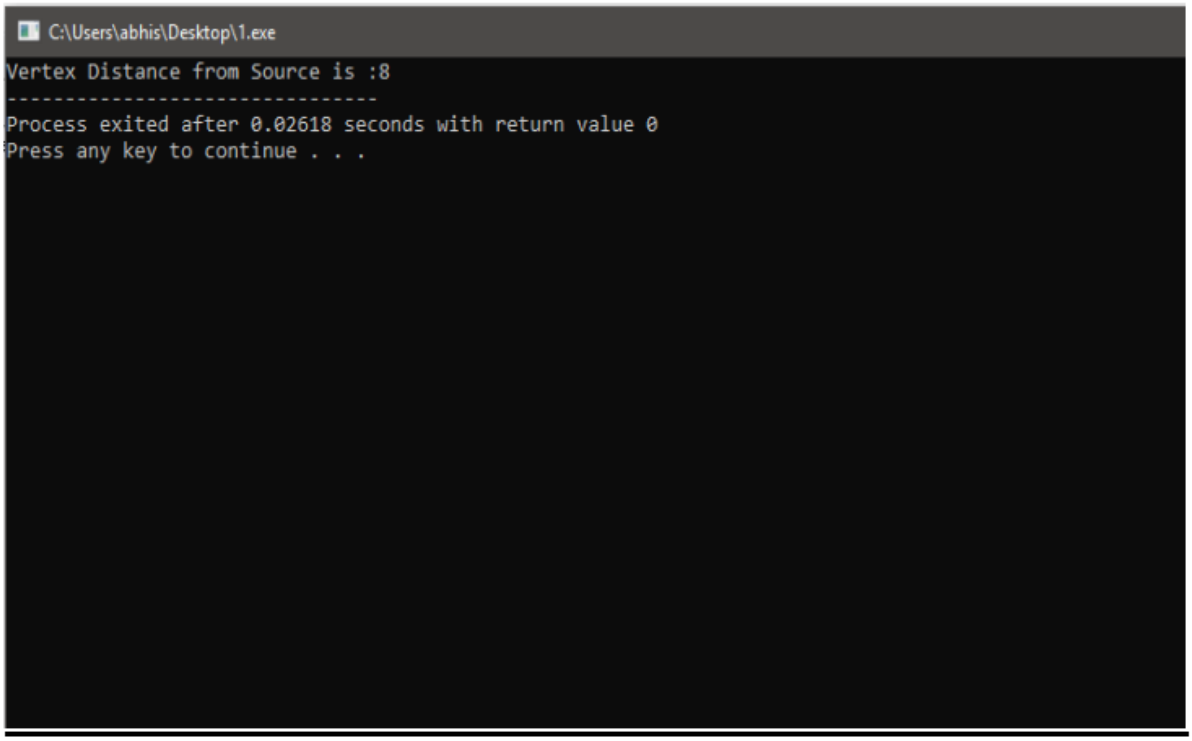




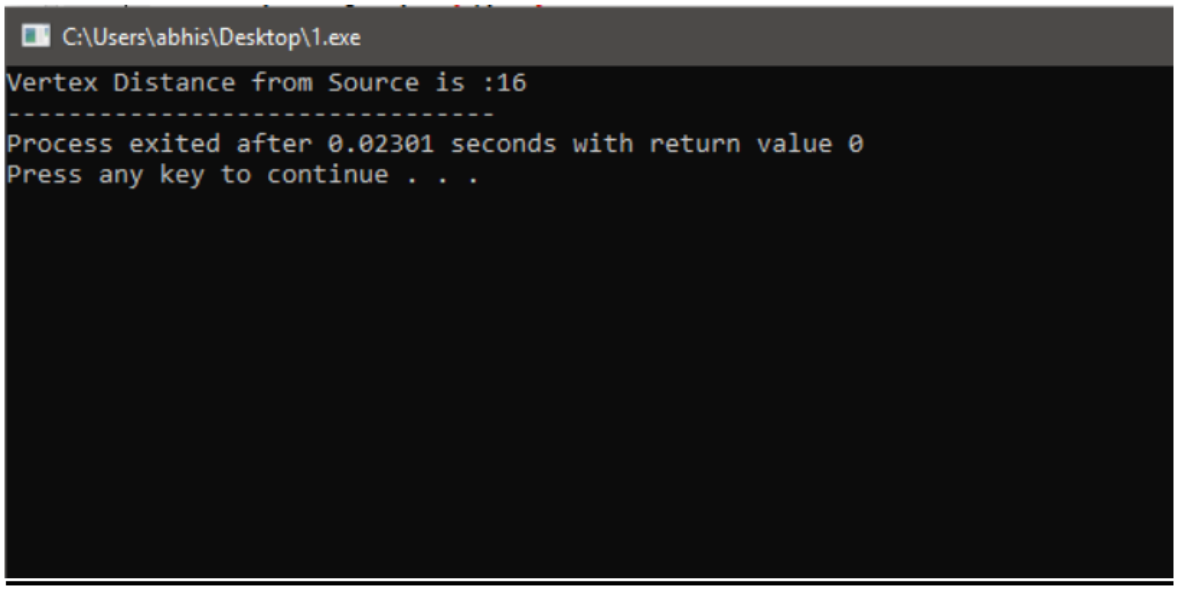
2. A-Star Algorithm


C:\Users\abhis\Desktop\1.exe
Vertex Distance from Source is :5

Process exited after 0.02604 seconds with return value 0
Press any key to continue . . .

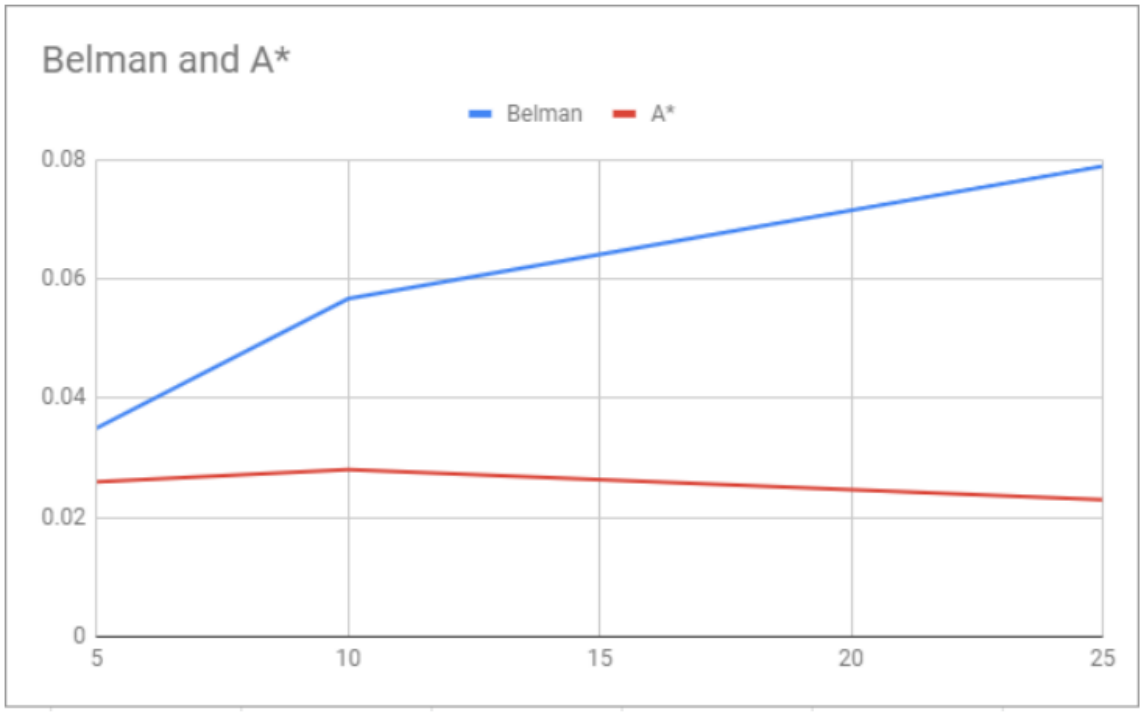

C:\Users\abhis\Desktop\1.exe
Vertex Distance from Source is :8

Process exited after 0.02618 seconds with return value 0
Press any key to continue . . .


C:\Users\abhis\Desktop\1.exe
Vertex Distance from Source is :16

Process exited after 0.02301 seconds with return value 0
Press any key to continue . . .

Result



We have thus compared both the algoritms by plotting the graphs of the time complexities takenby both the algorithms.

Inference

From the graph we infer that as n grows the bellman ford algorithm which uses dynamic programming, the time complexity also grows exponentially while for the A star algorithm which uses heuristics the time complexity remains nearly constant.This is because the astar algorthm does not visit every node and finds the minimum cost from source to destination by using the approximate heuristic function available for every node.

On the other hand the bellman ford algorithm which uses dynamic programming visits every node and also provide the minimum cost from source to every other node.

That is why dynamic programing gives the most optimal solution but the heuristic technique might not give the best optimal solution but it consumes less time than dynamic programing.

References

A Star Algorithm base paper

https://pdfs.semanticscholar.org/e2fc/bf5b2167c0c14d1be708e9a78b18616da474.pdf

Bellman Ford Algorithm base paper

https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.ijedr.org/papers/IJEDR1801130.pdf&ved=2ahUKEwi7ybvC08LhAhWLerwKHVsdDvsQFjADegQIARAB&usg=AOvVaw0t1IhfdHQ_r8kfMjRRU730