**UMT**

**Advanced Algorithms and Programming**

**Quantifying greedy strategies’ efficiencies by statistical validation procedure.**

**Worked by: Eljon Zagradi Professor: Rene Natowicz**

*I conducted testing with 2000 instances for each algorithm in my program. This large sample size helped me analyze the performance differences between the greedy and dynamic programming approaches comprehensively.*

*For each of the problems I have provided “randomArray” function which generates a random array of integers 1D or 2D depending on the Problem. Also I provided the “getR” which calculates the relative distace of greedy and dynamic programming in (%). If needed also the display functions are provided.*

*For each Problem is provided a program that tests and compares the performance of the dynamic programming and greedy algorithms. The program conducts 2000 iterations, generating random instances of the problem and measuring the execution times for both algorithms. It calculates the maximum -> (MVB, MSM, TB) or minimum -> (MCP) value obtained by each algorithm and collects the results. Finally, the results are visualized in a histogram, providing insights into the performance differences between the two approaches. The code demonstrates a systematic approach to evaluating and comparing algorithmic solutions for all the problems.*

*The printouts and explanation for each problem are saved at respective files:*

*Maxiimum Value Bag -> MVB.txt*

*Maximum Sum of Marks -> MSM.txt*

*Minimum Cost Path -> MCP.txt*

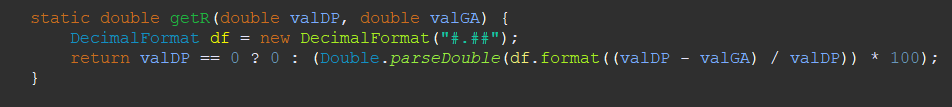
*Two Bags - > TB.txt*

*The full code for the Project is provided in this email but also can be found here: https://github.com/eljonzagradi/UMT\_Eljon\_Zagradi*

1. Maximum Value Bag Algorithm

A screen shot of a computer code

Description automatically generated with low confidence



A screen shot of a computer program

Description automatically generated with low confidence

A screen shot of a computer code

Description automatically generated with low confidenceA screen shot of a computer program

Description automatically generated with low confidence

A picture containing text, screenshot, software

Description automatically generated

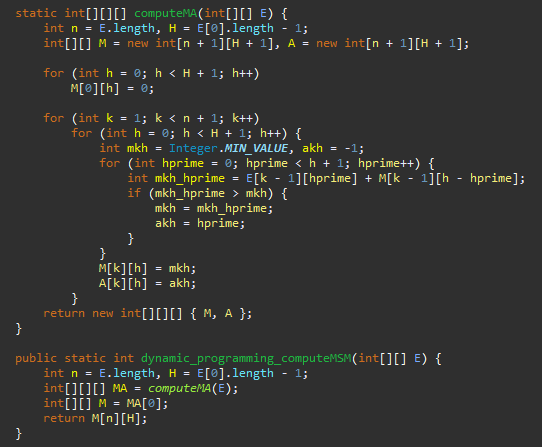
1. Maximum Sum of Marks:

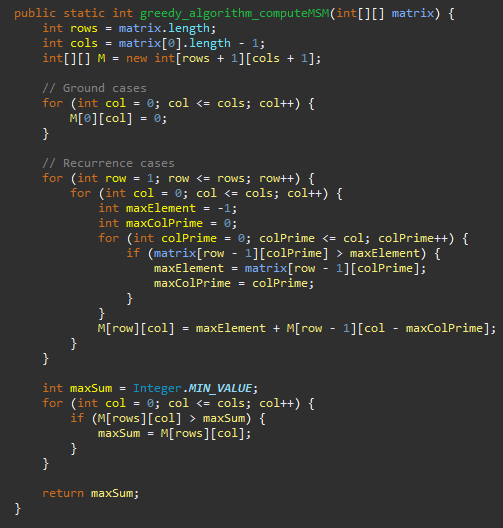
A screen shot of a computer program

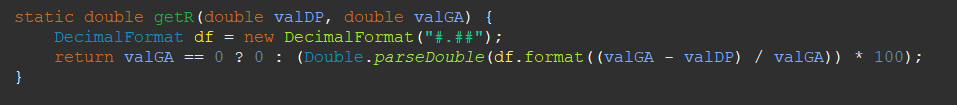
Description automatically generated with low confidenceA picture containing text, screenshot, font

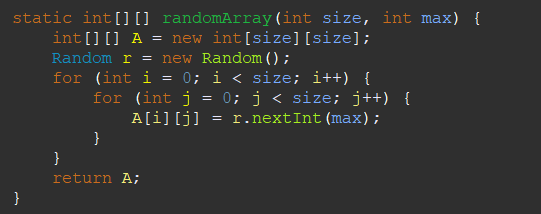
Description automatically generatedA picture containing screenshot, text, font

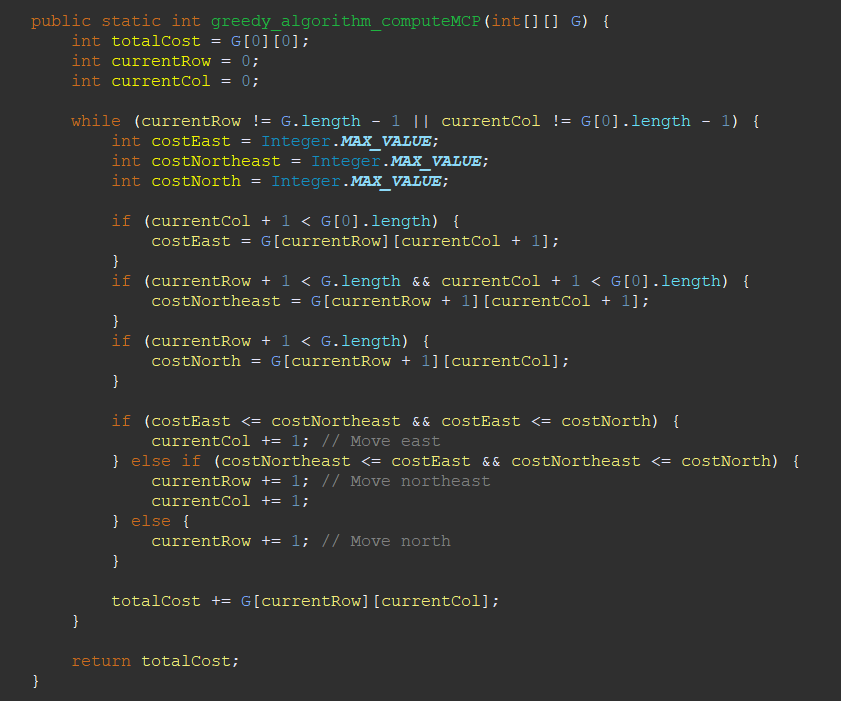
Description automatically generated





1. A picture containing text, screenshot, font

   Description automatically generatedMinimum Cost Path



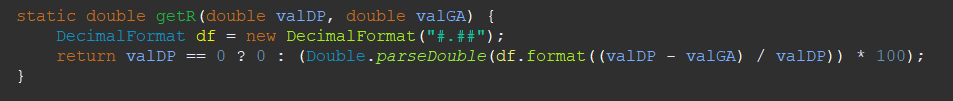
A picture containing text, screenshot

Description automatically generated

1. A screen shot of a computer

   Description automatically generated with medium confidenceA picture containing text, screenshot, font

   Description automatically generatedA screen shot of a computer program

   Description automatically generated with low confidenceTwo Bags

A screen shot of a computer program

Description automatically generated with low confidence

A screen shot of a computer program

Description automatically generated with low confidence

A screen shot of a computer code

Description automatically generated with low confidence

A screen shot of a computer code

Description automatically generated with low confidence