

Project 2 - Comparing Exhaustive Search with Greedy Algorithms

This project is due **Tuesday, April 19 at 11:59PM**. Upload a zipped file named `yourlastnameFirstinitial_proj2` to Blackboard. The file must include a(n):

1. Executive Summary Report in PDF format
2. Python (`.py`) file
3. `readme.txt` file with instructions on how to run your code

You are a high-level security manager at the Gombel Security Firm. On any day you have access to n professional cryptographers and m messages to crack. Today, you have 4 hackers at your disposal and 4 intercepted messages that need to be decrypted immediately.

Each cryptographer can only be assigned to exactly one message and exactly one message can be assigned to one cryptographer. Your goal is to find an assignment that minimizes the total cost to crack the codes. The cost matrix is below:

	MSG1	MSG2	MSG3	MSG4
Jill	9	2	7	8
Sven	6	4	3	7
Bud	5	8	1	8
Kevin	7	6	9	4

Although you are given this small example, your code should be able to read in any n by m matrix where n and m are equal. Here are the steps to complete the project:

- Step 1. Code an exhaustive search algorithm to find the optimal solution to the above problem.
- Step 2. Code a greedy algorithm to find the optimal solution to the problem.
- Step 3. Answer the following questions:
 1. What is the time efficiency of your exhaustive search algorithm and your greedy algorithm?
 2. Which works better for large n by m inputs?
 3. Does your greedy algorithm always yield an optimal solution for any matrix size n by m ? Supply a proof that either proves or disproves your answer.

- Step 4. Write your professional report that includes an executive summary and answers the above questions.

Grading Rubric

1. Implementation (**70 points**)

- (a) Execution. When the code is run it reads in a matrix from a file and performs both the exhaustive search and greedy algorithms back to back. The code works for any n by m matrix.
- (b) Heavily Documented
 - i. A header block that contains your name, assignment, brief description of the code
 - ii. Comments throughout the code to help the grader understand your thought process
 - iii. Your code comes with a readme file with instructions on how to run your code and use the menu-driven interface

2. Executive Summary Report (**30 points**). Your report contains the following:

- (a) Executive Summary
- (b) Pseudocode for both algorithms
- (c) Answers to the questions
- (d) Form and Style. Grammatically correct with no spelling errors, easy to read and understand