Paint Shop Programming Challenge

### By: Mina Youssef

### Date: 19/10/2018

# Introduction

This document reports on Mina Youssef solution of paint shop programming challenge requested by HBC as part of recruitment process

GitHub location: <https://github.com/minavyoussef/HBC-PaintShop.git>

Language/IDE: Python 3+/PyCharm

# Methodology

I’ve implemented two approaches in solving the given challenge, one by Brute Force and other using elimination

## Brute Force Approach

In this approach I regarded the solution space as data points:

cheapest option : [('G', 'G', 'G', 'G', 'G') for color\_count = 5]

most expensive option: ('M', 'M', 'M', 'M', 'M') for color\_count = 5

Search Space:

'G', 'G', 'G', 'G', 'G'

'G', 'G', 'G', 'G', 'M'

'G', 'G', 'G', 'M', 'G'

.

.

'M', 'M', 'M', 'M', 'M'

And for every solution I check if it satisfies all customers, if non-exists return ‘No Solution found’

Code: ~\HBC\solvers\BruteForceOptimisation.py

## Elimination Approach

In this approach start by cheapest paint combination [('G', 'G', 'G', 'G', 'G') for color\_count = 5] and for every customer that doesn’t satisfy the combination I update the solution to confirm with the customer till all customers are satisfied and if not, I return ‘No Solution found’

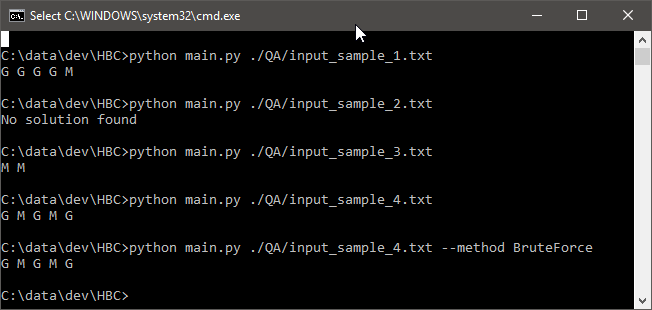
Code: ~\HBC\solvers\EliminationOptimisation.py

# Execution

Navigate to HBC directory

Open cmd

$> python main.py <input\_file>



Options –method [BruteForce | Elimination]

# QA

I’ve create a unittest for both methods under ~\HBC\main\_unittest.py.

Also, I’ve created set of test cases varies from simple to complex under ~\HBC\QA\\*.txt

