CS 455 - Artificial Intelligence
Module 2 Homework - Genetic Algorithms
Cameron Stark & Dustin Cribbs

Problem 1:

i. 11 01 10 11
- r1 = Green
- r2 = White
- r3 = Blue
- r4 = Green
- fitness = 2 + 3 + 3 + 2 = 10

ii. 00 01 00 11
- r1 = Red
- r2 = White
- r3 = Red
- r4 = Green
- fitness = 1 + 3 + 2 + 2 = 8

iii. 11 11 11 10
- r1 = Green
- r2 = Green
- r3 = Green
- r4 = Blue
- fitness = 0 + 1 + 1 + 2 = 4

iv. 01 11 10 10
- r1 = White
- r2 = Green
- r3 = Blue
- r4 = Blue
- fitness = 2 + 3 + 2 + 1 = 8

Chromosome 1: 11011011
Chromosome 2: 00010011
New Chromosome 1: 11010011
New Chromosome 2: 00011011

11 01 00 11
- r1 = Green
- r2 = White
- r3 = Red
- r4 = Green
- Fitness(new Chromosome 1): 2 + 3 + 3 + 2 = 10

00 01 10 11

r1 = Red
r2 = White
r3 = Blue
r4 = Green
Fitness(new Chromosome 2): 2 + 3 + 3 + 2 = 10

Chromosome 3: 01100011
New Chromosome 3: 01101011
01 10 10 11

r1 = White
r2 = Blue
r3 = Blue
r4 = Green
Fitness(new Chromosome 3): 2 + 2 + 2 + 2 = 8

Problem 2:

a. To prevent a local search from getting stuck in some sort of local maxima or minima, the implementation of crossover or mutating can be used to give the search a new location or direction to be searching in, crossover is the action of taking values (which are the two with the highest fitness) and exchanging aspects of each part with the other and then continuing from that new value which does not guarantee a solution but rather a potential way to get out of the local point. The other method is mutation which takes the value (maybe the highest fitness) and changes an aspect of it (in the case of a binary value a bit is flipped) this can possibly push the search in another direction to get it out of the local point.

b. Mutation is important because it allows the search algorithm to randomly change position allowing for a new approach to the search, and is not reliant on the mating of two other chromosomes.

c. true

d. True

Problem 3:

Task 1: The definition of the chromosome in this case a binary bit string with 0's indicating the item is not in the specific bag and 1's indicating the item is in the bag. For example:

Weights = [5, 10, 15, 20, 25]
Profits = [3, 6, 9, 12, 15]
Chromosome = [0 1 0 1 0]

Which will have a knapsack weight of 30 and profit of 18. The chromosome itself will be determined by what the knapsack max weight is.

As for the implementation of a multi-knapsack situation the chromosomes would be changed from a 0 and 1 to the knapsack id that the item is in that knapsack, this is so that an item does not exist in multiple knapsacks as stated in the problem statement.
For Example:

Weights = [5, 10, 15, 20, 25]
Profits = [3, 6, 9, 12, 15]
Knapsacks = [0, 1, 2, 3]
Chromosome = [0, 3, 2, 1, 1]

Task 2: The fitness function for this is the sum of the profits in each knapsack with an overflow of the “weight remaining'' in a knapsack making the fitness 0, this was chosen to not allow the poor fitness chromosomes to be selected for reproduction.