1. (a) S'={1,2,3}

S={7}

(b) S'={1}

S={1000}

(c) To find an optimal solution using genetic algorithms and bit strings you would be best off adding together the subsets to get the sum of that number in the subset. You would then convert the sum to binary and compare the subsets using the binary genetic algorithm.

For example:

S={3,5,8,14,30} is the original subset, the optimal solution is:

S'={3,5,8,14}

To make this verifiable we would add the subset together. 3+5+8+14 = 30 and do the same for the numbers in the original subset 30=30.

Convert both subsets to binary...

S= 30 → 11110

S'= 30 → 11110 Both are the same so the solution works!

(d) To calculate the fitness the program would have to generate its subsets and convert them to binary in the same manner that is described above. Once this is done, the fitness will need to be calculated. The simplest way to do this would be a comparison of the two bit strings bit by bit. For each bit that matches between the two strings, one point will be added to the fitness and the highest fitness number would be considered the optimal solution.

2.

A description...

1. (a) To get an output of -1, x and y will have to be the following:

x = -1

y = 2

(b) To get an output of 1, x and y will have to be the following:

x = 4

y = 3

(c) w1 = 3

w'1 = 3 + .5(1+1) -1 = -4

**w'1 = -4**

w2 = -1

w'2 = -1 + .5(1+1) 0 = 0

**w'2 = 0**

1. (a) under(X,Y) :-

on(X,Y).

(b) second(X) :-

on(X,Y),

on(Y,Z).

(c) third (X) :-

on(X,Y),

on(Y,Z),

on(Z,A).

1. After the hopfield network “settles down” into a stable state I would expect the values at the six nodes to be **(-1, -1, -1, 1, 1, 1)**.
2. (a) After tabulating which category predicted whether or not the person would eat at the restaurant it turned out that the price was the biggest factor. In 4 out of 5 cases where the good was inexpensive the student ate there. Not once did the student eat at an expensive restaurant.

(b)

A description...

(c)

A description...

7. (a) The number of moves necessary