Genetic Algorithms Assignment #1 (5%)

Instructor: Tian-Li Yu

1. Let's check how often deception occurs in a random function. The definition of deception can be found in the lecture slides.
   (a) Consider the case with two genes. By assigning fitness values for $f(00)$, $f(01)$, $f(10)$, and $f(11)$, does deception ever occur? Prove it cannot happen if not, give an example if yes.
   (b) Consider the case with three genes, randomly assign the fitness values for $f(000)$, $f(001)$, $f(010)$, $f(011)$, $f(100)$, $f(101)$, $f(110)$, and $f(111)$ with uniform distribution from 0 to 1. Repeat the experiments 10^6 times. What's the probability that 3-deception occurs?
      - For example, if $f(011)$ is the greatest, 3-deception occurs **iff** $f(1**) > f(0**)$; $f(*0*) > f(*1*)$; $f(**0) > f(**1)$; $f(10*) > f(00*), f(01*), f(11*)$; $f(1*0) > f(0*0), f(0*1), f(1*1)$; $f(*00) > f(*01), f(*10), f(*11)$.
   (c) Repeat (b), but with 4 genes.
   (d) For a problem with ell genes (problem size), the probability that k-deception does **NOT** occur among any k genes is roughly $(1-p)^{C_k^{ell}}$, where p is what you recorded in (b) and (c). What's the problem size that makes 3-deception occur with probably 0.5? What's that for 4-deception? When does 3-deception occur more often than 4-deception or the other way around? Write a short essay of your finding.

2. Thierens' convergence-time model assumes perfect mixing. Let's verify the model with one-point XO, uniform XO, and population-wise shuffling. In population-wise shuffling, the set of genes at a particular position of offspring is a random shuffle of the set of genes at the same position of parents. (Use tournament selection of size 2 without replacement and no mutation.)
   (a) Experiment these three XO's with a SGA on the OneMax problem with different sizes (ell) of 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500. Plot the results on a figure with problem size versus convergence time. (You need to average over at least 30 independent runs to get stable results. Population size should be set at $4*ell*\ln(ell)$.)
   (b) What's the theoretical value of selection intensity? How does that compare with the selection intensity measured in your experiments? Is selection intensity really invariant with generation?
   (c) How does Thierens' model compare with your results? Write a short essay of your finding.

- You may use the SGA code released on ceiba. You'll need to implement population-wise shuffling. MyRand::uniformArray() may come in handy. To generate a random shuffle array of 0~n-1. Simply use the following code:

```
int *A = new int[n];
myRand.uniformArray(A, n, 0, n-1);
```

  Also be sure to check the following parts:

  (1) Chromosome::getMaxFitness() to return (length-1e-6);

  (2) Chromosome::evaluate() to return oneMax();

  (3) SGA::shouldTerminate() to check if population[0].getMaxFitness() <= stFitness.getMean();

Note:

(1) Use any programming language as you like.

(2) Do **NOT** hand in source codes.

(3) Submit electronically on ceiba by the due date.