Members:

Registration Id	A01746886
Registration Id	A01746887

Instructions: Carefully read the information and answer according to the request.

Local Search

1. Suppose you want to MAXIMIZE the following objective function:

$$obFunc = 9a + bc + cd + 5e + 2$$

where a-e are all boolean variables (0's and 1's) representing a solution of the problem and you must toggle one variable value of the current solution (*c:a b c d e*) at the time to get the neighbors of the current solution (*n:a b c d e*).

a. Manually run up to 3 iterations of the Hill Climbing algorithm from the initial state:

N_i	*c:a b c d e*	*obFunc*	*n: a b c d e*	*obFunc*
1.	0 1 1 0 0	3	1. 1 1 1 0 0	12
			2. 0 0 1 0 0	2
			3. 0 1 0 0 0	2
			4. 0 1 1 1 0	4
			5. 0 1 1 0 1	8
2.	1 1 1 0 0	12	1. 0 1 1 0 0	3
			2. 1 0 1 0 0	11
			3. 1 1 0 0 0	11
			4. 1 1 1 1 0	13
			5. 1 1 1 0 1	17
3.	1 1 1 0 1	17	1. 0 1 1 0 1	8
			2. 1 0 1 0 1	16
			3. 1 1 0 0 1	16
			4. 1 1 1 1 1	18
			5. 1 1 1 0 0	12

Best solution = 1 1 1 1 1 **Objective value =** 18

b. Manually run up 7 iterations of the SIMULATED ANNEALING algorithm now to MINIMIZE the value of the objective function of the above problem from the indicated initial current solution (*c:a b c d e*), using the random numbers (*rand-bit*) shown that indicate the bit to toggle for getting successor solutions (*s:a b c d e*), and the random numbers (*rand#*) shown that are used to verify the eligibility for the successor, starting at the temperature T=60 with a cooling rate of 10. You must not compute the acceptance probability (*accept-prob*) of improving successors.

N_i	c:a b c d e	obFunc	rand-bit	s: a b c d e	obFunc	Temp	Accept-Prob	rand#	
1.	11010	11	2	10010	11	60	$e^{(0/60)}=1$	0.62	ΔE = 0
2.	10010	11	4	10000	11	50	$e^{(0/50)}=1$	0.17	ΔE = 0
3.	10000	11	5	10001	16	40	1	0.43	ΔE = 5
4.	10001	16	3	10101	16	30	$e^{(0/30)}=1$	0.23	ΔE = 0
5.	10101	16	2	11101	19	20	1	0.51	ΔE = 1
6.	11101	19	1	01101	8	10	$e^{(-9/10)}=0.40$	0.02	
7.	01101	8	3	01001	7	0	$e^{(-1/0)}=$ NC	0.80	

Best solution = 01001 **Objective value =** 7 error

Genetic algorithms

2. Suppose you want to find the individual that MAXIMIZES the fitness function:

$$fitness = -\frac{1}{1000}x(x-12)(x-20)(x-28)+20$$

Where x is the variable to be determined in the range of 0 to 31, which must be encoded on a chromosome of 5 bits. Calculate the fitness of each of the members of the initial population, below. Also calculate the probability of each individual to be selected during a process of fitness-proportionate reproduction.

	Initial Population			*Reproduction*	***Cumulative***
N_i	*x*	*chromosome*	*fitness*	*probability*	***probability***
1.	01	00001	25.643	0.33 (0.326)	0.33
2.	10	01010	23.600	0.30	0.63
3.	20	10100	20.000	0.25 (0.254)	0.88
4.	30	11110	9.200	0.12	1.00

Population's total fitness: 78.443 **Average fitness:** 19.6107

Assume the sequence of random numbers (r#) given, and use them to simulate sexual reproduction (crossover) process between the individuals in the population, use the random numbers to select parents and to define the random crossover point (after the indicated bit).

Selection and crossover:

	Parent1		Parent2		Cross-point	Offspring (just the 1st.)		
Ni	r#	chromosome	r#	chromosome	r#	chromosome	individual	fitness
1.	0.30	00001	0.75	10100	3	00000	00	20.000
2.	0.63	01010	0.80	10100	1	00100	04	32.080
3.	0.91	11110	0.50	01010	4	11110	30	9.200
4.	0.03	00001	0.39	01010	2	00010	02	29.36

Again, assume the given sequence of random numbers (r#) and use them to simulate possible mutations of 1 bit in the offspring with a mutation probability of 0.2. The first number mutates and decides whether the second bit to mutate.

Mutation:

					Final Population		
Ni	offspring	r#	Mutates?(Yes/No)	r#Bit	chromosome	individual	fitness
1.	00000	0.80	NO	1	00000	00	20.000
2.	00100	0.10	Yes	5	00101	05	32.075
3.	11110	0.32	NO	3	11110	30	9.200
4.	00010	0.05	Yes	4	00000	00	20.000

Population total fitness: 81.275 **Average fitness:** 20.3187

if r# <= 0.20 mutate
else
equal chromosome