HW۲

محمدرضا صاحب زاده

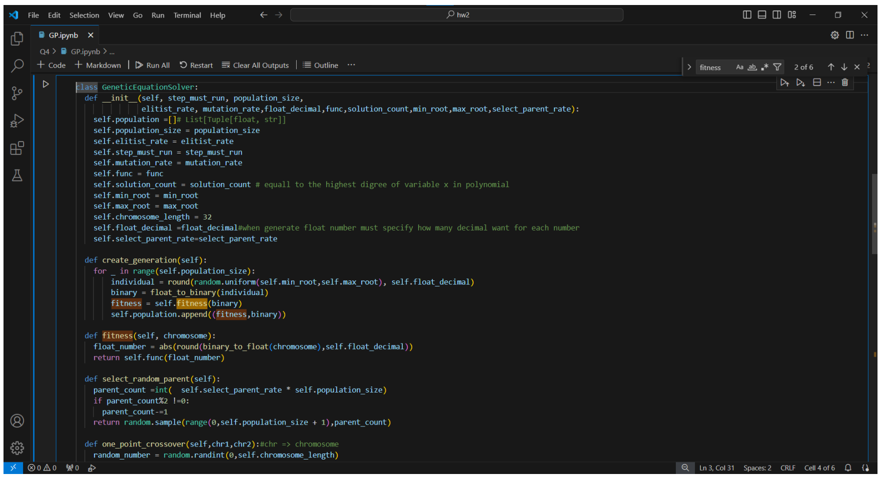
۲۶ خرداد ۱۴۰۳

دانشگاه علم و صنعت ایران

فهرست مطالب

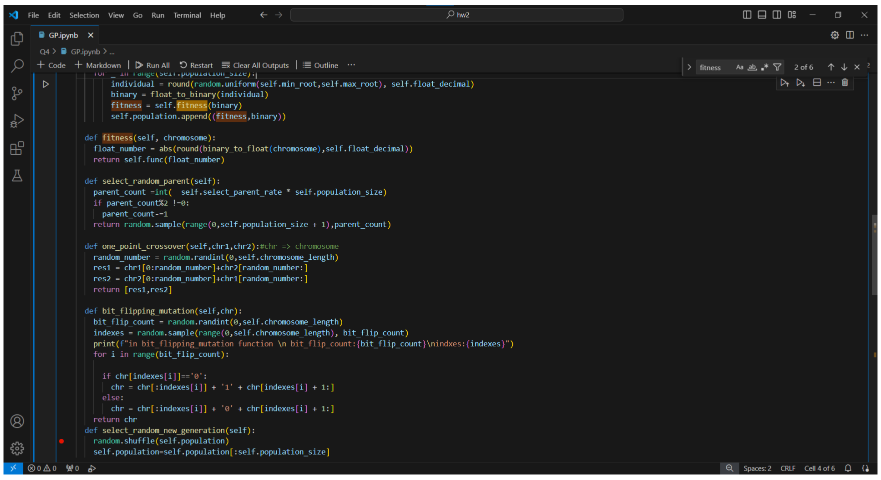
Root finder Genetic

در مراحل اول بیشتر الگوریتم ها تصادفی هست و تنظیمات اولیه در دو تصویر زیر معلوم هست



```python
class GeneticEquationSolver:
  def __init__(self, step_must_run, population_size,
               elitist_rate, mutation_rate,float_decimal,func,solution_count,min_root,max_root,select_parent_rate):
    self.population =[]# List[Tuple[float, str]]
    self.population_size = population_size
    self.elitist_rate = elitist_rate
    self.step_must_run = step_must_run
    self.mutation_rate = mutation_rate
    self.func = func
    self.solution_count = solution_count # equall to the highest digree of variable x in polynomial
    self.min_root = min_root
    self.max_root = max_root
    self.chromosome_length = 32
    self.float_decimal =float_decimal#when generate float number must specify how many decimal want for each number
    self.select_parent_rate=select_parent_rate

  def create_generation(self):
    for _ in range(self.population_size):
        individual = round(random.uniform(self.min_root,self.max_root), self.float_decimal)
        binary = float_to_binary(individual)
        fitness = self.fitness(binary)
        self.population.append((fitness,binary))

  def fitness(self, chromosome):
    float_number = abs(round(binary_to_float(chromosome),self.float_decimal))
    return self.func(float_number)

  def select_random_parent(self):
    parent_count =int(  self.select_parent_rate * self.population_size)
    if parent_count%2 !=0:
      parent_count-=1
    return random.sample(range(0,self.population_size + 1),parent_count)

  def one_point_crossover(self,chr1,chr2):#chr => chromosome
    random_number = random.randint(0,self.chromosome_length)
```

شکل ۱:

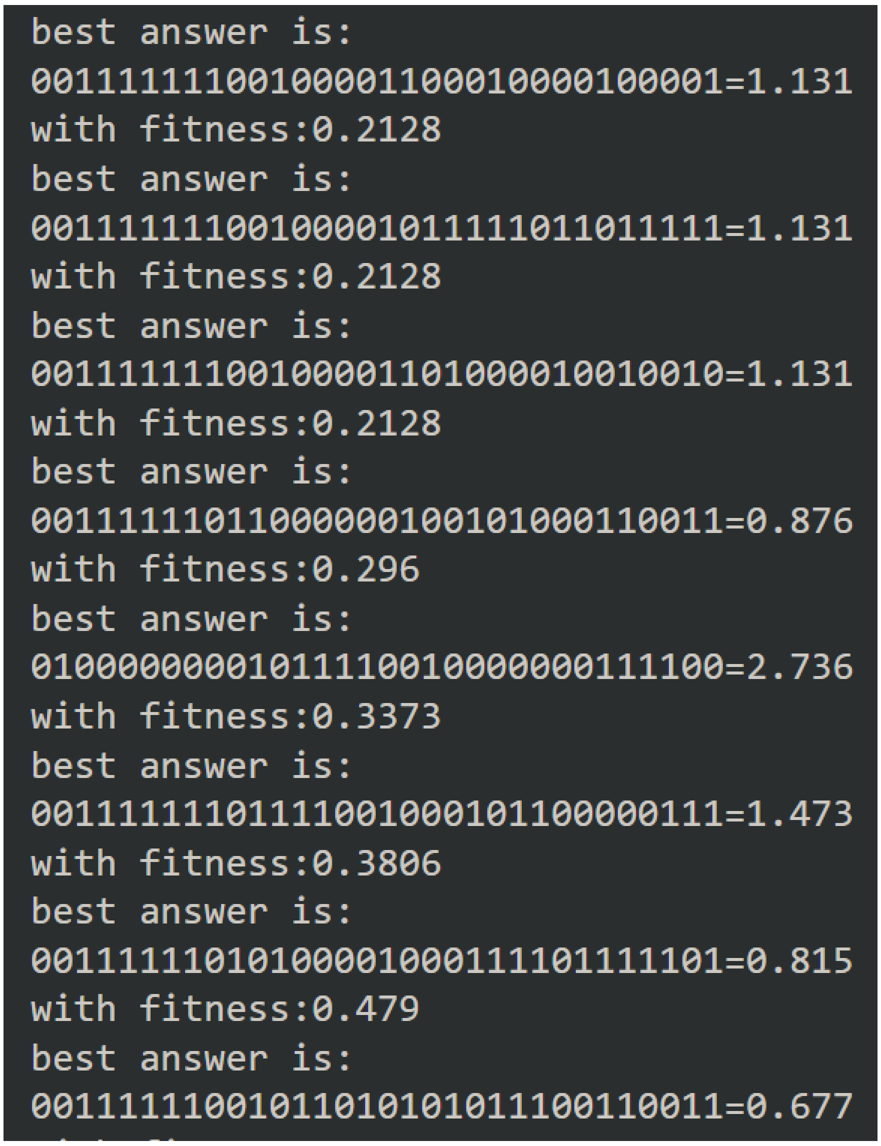


```python
        individual = round(random.uniform(self.min_root,self.max_root), self.float_decimal)
        binary = float_to_binary(individual)
        fitness = self.fitness(binary)
        self.population.append((fitness,binary))

  def fitness(self, chromosome):
    float_number = abs(round(binary_to_float(chromosome),self.float_decimal))
    return self.func(float_number)

  def select_random_parent(self):
    parent_count =int(  self.select_parent_rate * self.population_size)
    if parent_count%2 !=0:
      parent_count-=1
    return random.sample(range(0,self.population_size + 1),parent_count)

  def one_point_crossover(self,chr1,chr2):#chr => chromosome
    random_number = random.randint(0,self.chromosome_length)
    res1 = chr1[0:random_number]+chr2[random_number:]
    res2 = chr2[0:random_number]+chr1[random_number:]
    return [res1,res2]

  def bit_flipping_mutation(self,chr):
    bit_flip_count = random.randint(0,self.chromosome_length)
    indexes = random.sample(range(0,self.chromosome_length), bit_flip_count)
    print(f"in bit_flipping_mutation function \n bit_flip_count:{bit_flip_count}\nindxes:{indexes}")
    for i in range(bit_flip_count):

      if chr[indexes[i]]=='0':
        chr = chr[:indexes[i]] + '1' + chr[indexes[i] + 1:]
      else:
        chr = chr[:indexes[i]] + '0' + chr[indexes[i] + 1:]
    return chr
  def select_random_new_generation(self):
    random.shuffle(self.population)
    self.population=self.population[:self.population_size]
```

شکل ۲:

نتایج زیر را برای هر پارامتر داریم

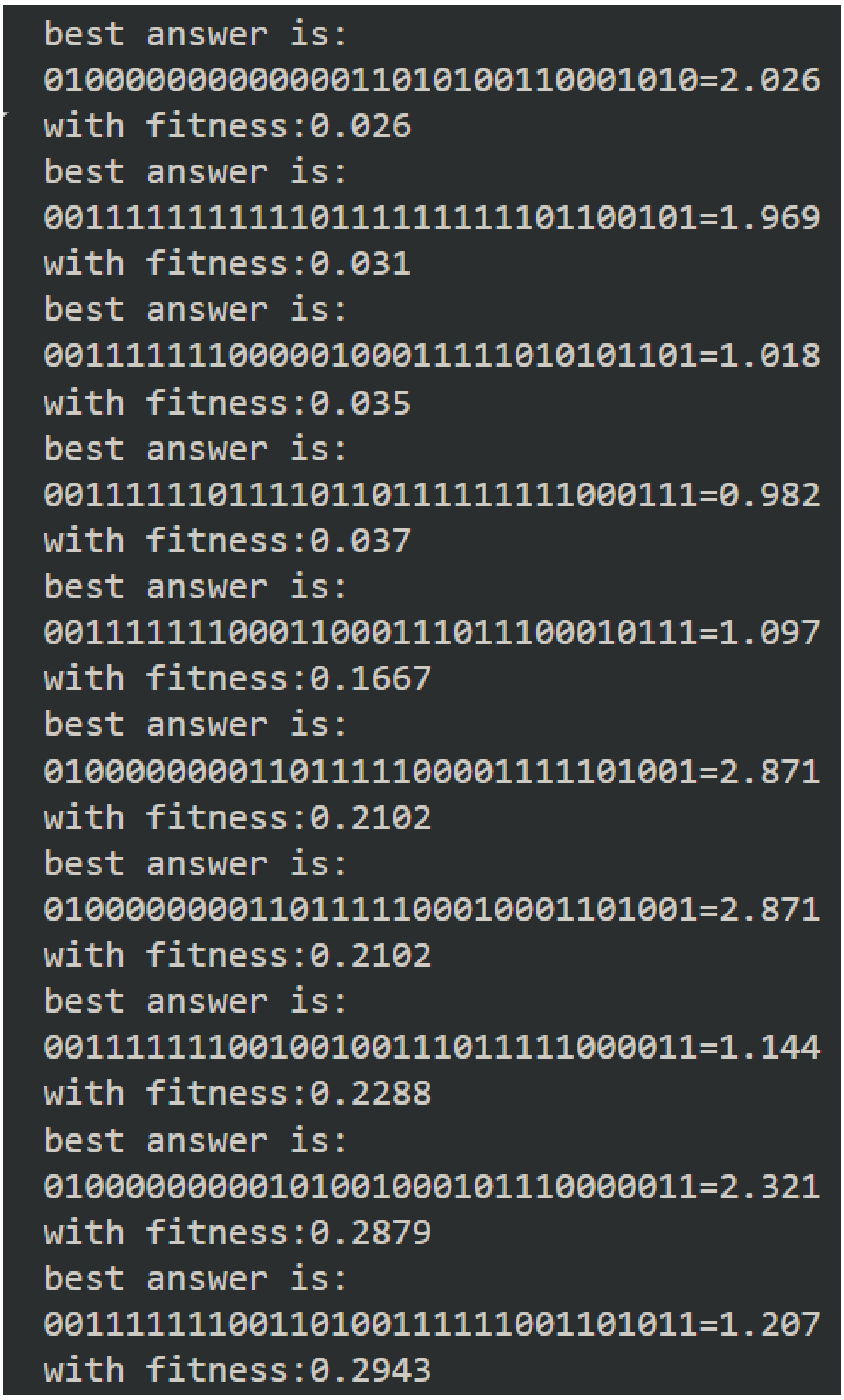
$$x**3-6*x**2+11*x-6$$



```
best answer is:
00111111100100001100010000100001=1.131
with fitness:0.2128
best answer is:
00111111100100001011111011011111=1.131
with fitness:0.2128
best answer is:
00111111100100001101000010010010=1.131
with fitness:0.2128
best answer is:
00111111011000000100101000110011=0.876
with fitness:0.296
best answer is:
01000000001011110010000000111100=2.736
with fitness:0.3373
best answer is:
00111111101111001000101100000111=1.473
with fitness:0.3806
best answer is:
00111111010100001000111101111101=0.815
with fitness:0.479
best answer is:
00111111001011010101011100110011=0.677
```

شکل ۳:

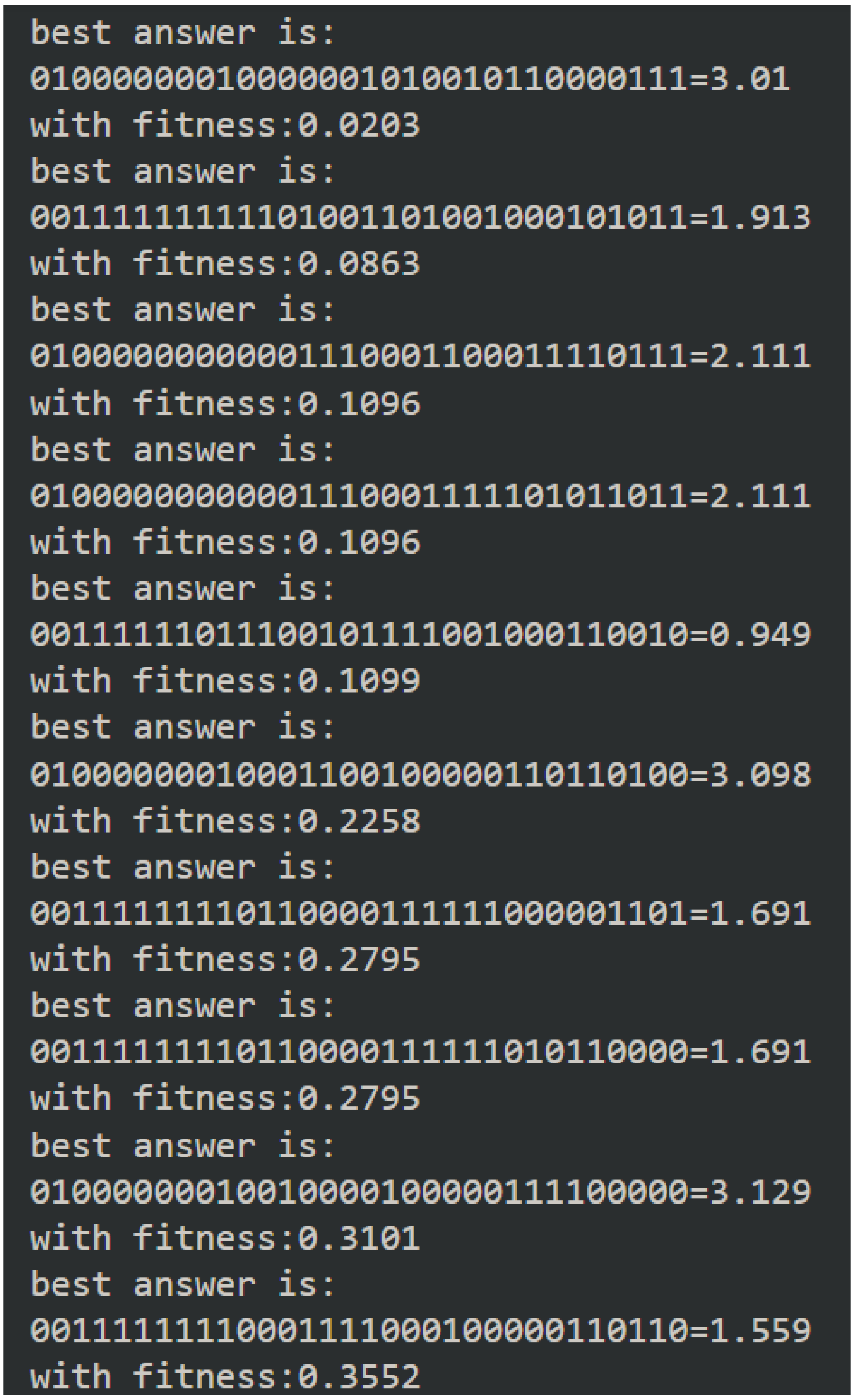
حال جمعیت اولیه را از ۱۰۰۰ به ۱۰۰۰۰ افزایش میدهیم



```
best answer is:
01000000000000001101010011000101 0=2.026
```

شکل ۴:

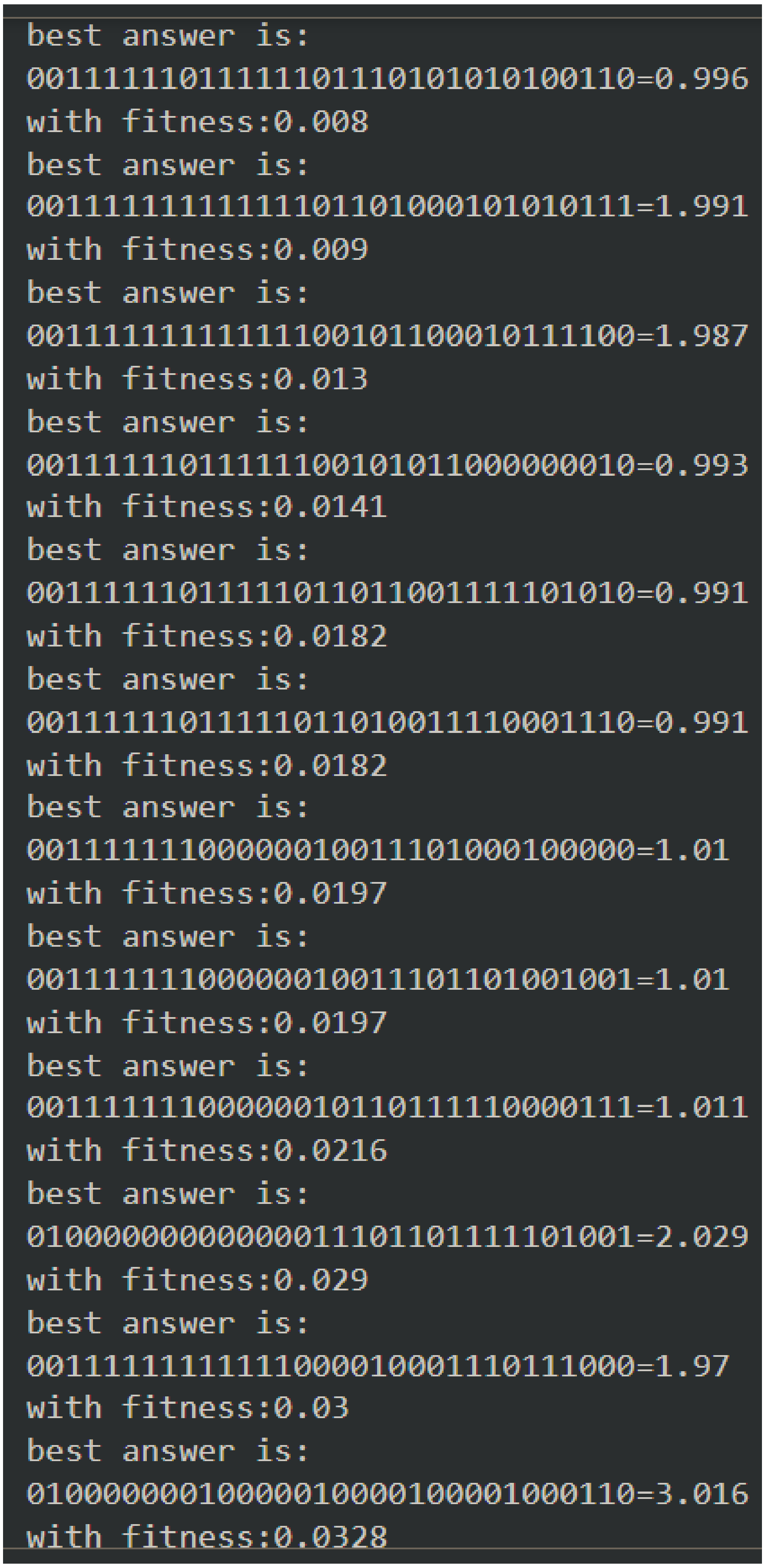
حال تعداد مراحل اولیه را از ۱۰۰ به ۱۰۰۰ افزایش میدهیم



```
best answer is:
01000000010000001010010110000111=3.01
with fitness:0.0203
best answer is:
0011111111101001101001000101011=1.913
with fitness:0.0863
best answer is:
01000000000001110001100011110111=2.111
with fitness:0.1096
best answer is:
01000000000001110001111101011011=2.111
with fitness:0.1096
best answer is:
00111111011100101111001000110010=0.949
with fitness:0.1099
best answer is:
01000000010001100100000110110100=3.098
with fitness:0.2258
best answer is:
00111111110110000111111000001101=1.691
with fitness:0.2795
best answer is:
00111111110110000111111010110000=1.691
with fitness:0.2795
best answer is:
01000000010010000100000111100000=3.129
with fitness:0.3101
best answer is:
00111111100011110001000000110110=1.559
with fitness:0.3552
```

شکل ۵:

میبینیم جواب های اولیه تقریبا دقیق هستند حال جمعیت اولیه را از ۱۰۰۰۰ به ۱۰۰۰۰۰ افزایش میدهیم و مینیمم و ماکسیمم را به -۵۰۰ و ۵۰۰ تغییر میدهیم

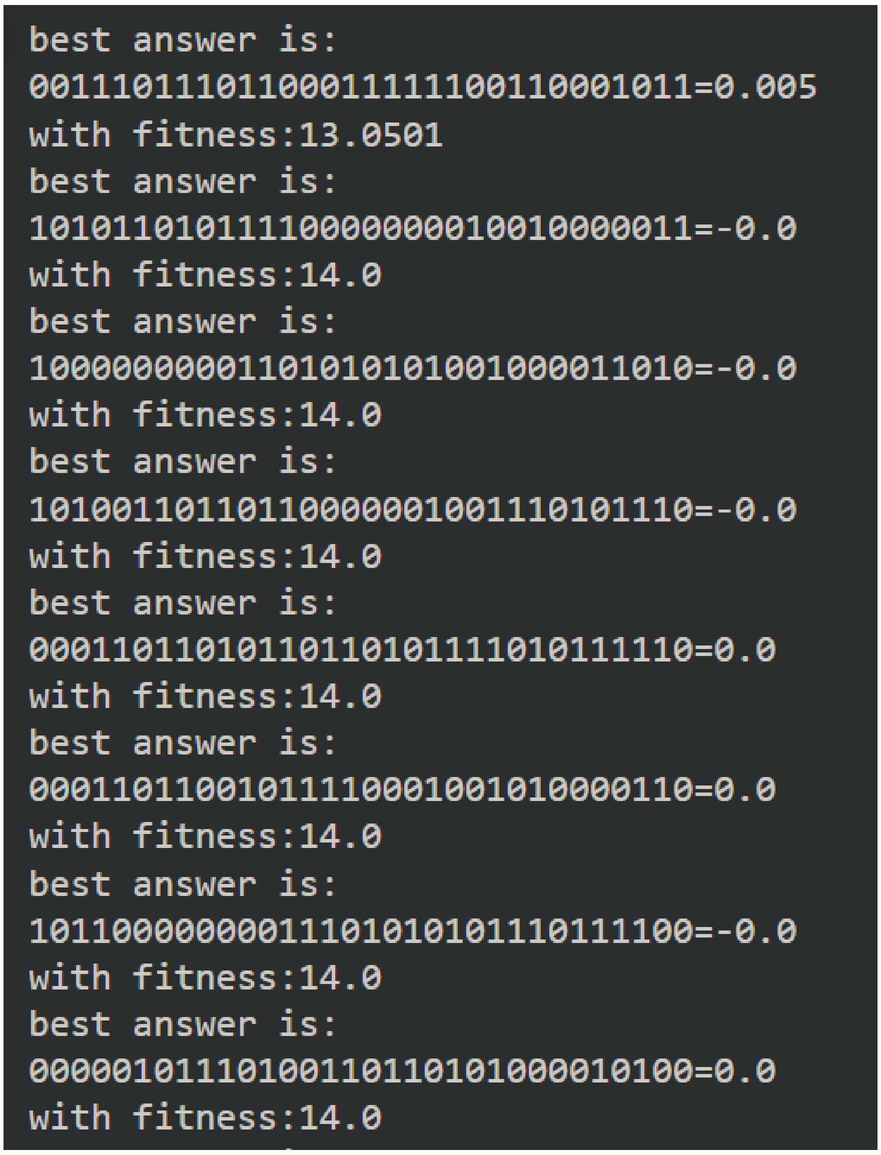


```
best answer is:
00111111011111101110101010100110=0.996
with fitness:0.008
best answer is:
00111111111111101101000101010111=1.991
with fitness:0.009
best answer is:
00111111111111100101100010111100=1.987
with fitness:0.013
best answer is:
00111111011111100101011000000010=0.993
with fitness:0.0141
best answer is:
00111111011111101101100111101010=0.991
with fitness:0.0182
best answer is:
00111111011111101101001111000110=0.991
with fitness:0.0182
best answer is:
00111111100000010011101000100000=1.01
with fitness:0.0197
best answer is:
00111111100000010011101101001001=1.01
with fitness:0.0197
best answer is:
00111111100000010110111110000111=1.011
with fitness:0.0216
best answer is:
01000000000000011101101111101001=2.029
with fitness:0.029
best answer is:
00111111111111000010001110111000=1.97
with fitness:0.03
best answer is:
01000000010000010000100001000110=3.016
with fitness:0.0328
```

شکل ۶:

نتایج زیر را برای معادله زیر و همان پارامتر های شکل ۳ و دو جواب ۰.۰۷ و ۴۷ داریم

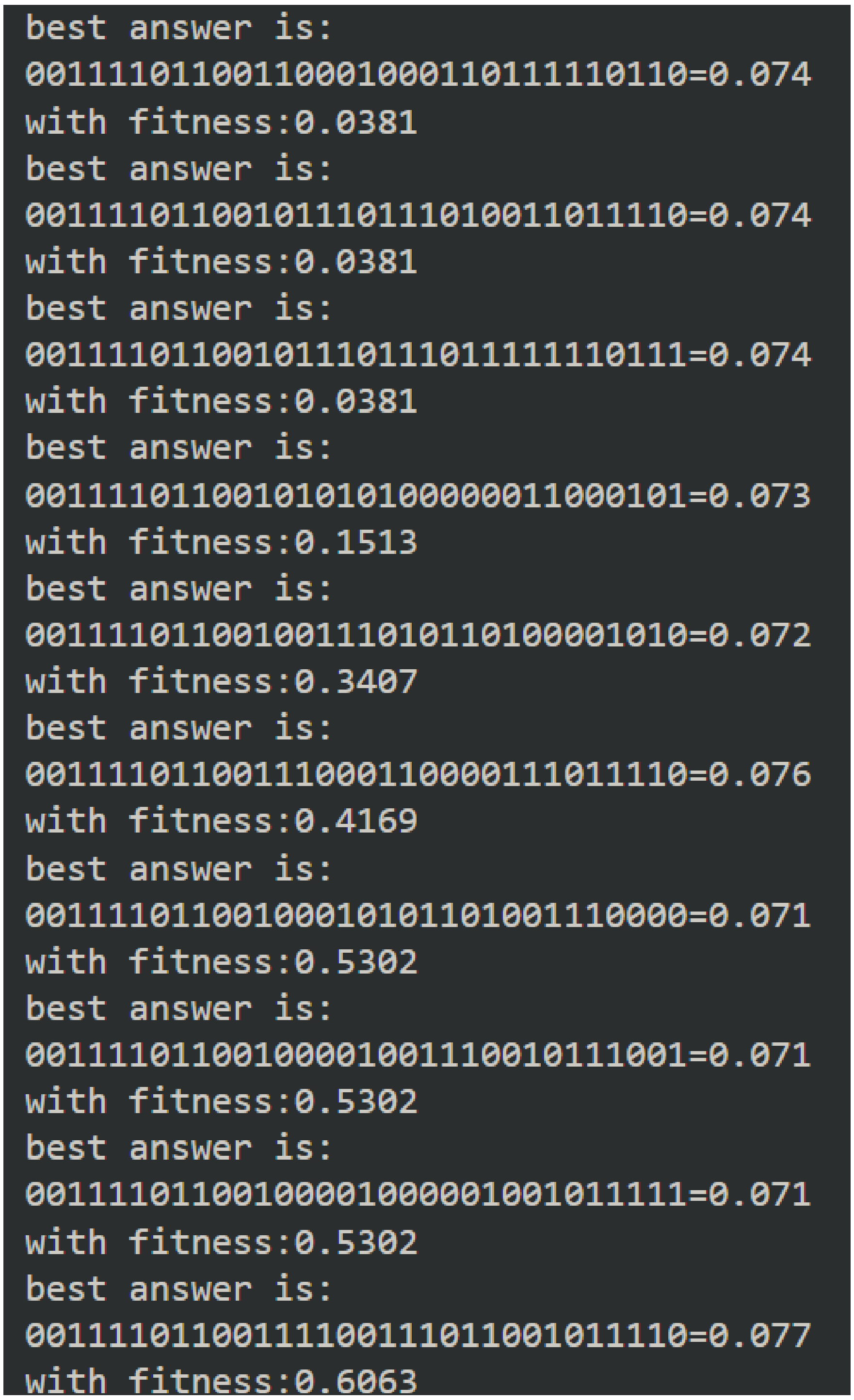
$4 * x * *2 - 190 * x + 14$



```
best answer is:
00111011101100011111100110001011=0.005
with fitness:13.0501
best answer is:
10101101011110000000010010000011=-0.0
with fitness:14.0
best answer is:
10000000001101010101001000011010=-0.0
with fitness:14.0
best answer is:
10100110110110000001001110101110=-0.0
with fitness:14.0
best answer is:
00011011010110110101111010111110=0.0
with fitness:14.0
best answer is:
00011011001011110001001010000110=0.0
with fitness:14.0
best answer is:
10110000000011101010101110111100=-0.0
with fitness:14.0
best answer is:
00000101110100110110101000010100=0.0
with fitness:14.0
```

شکل ۷:

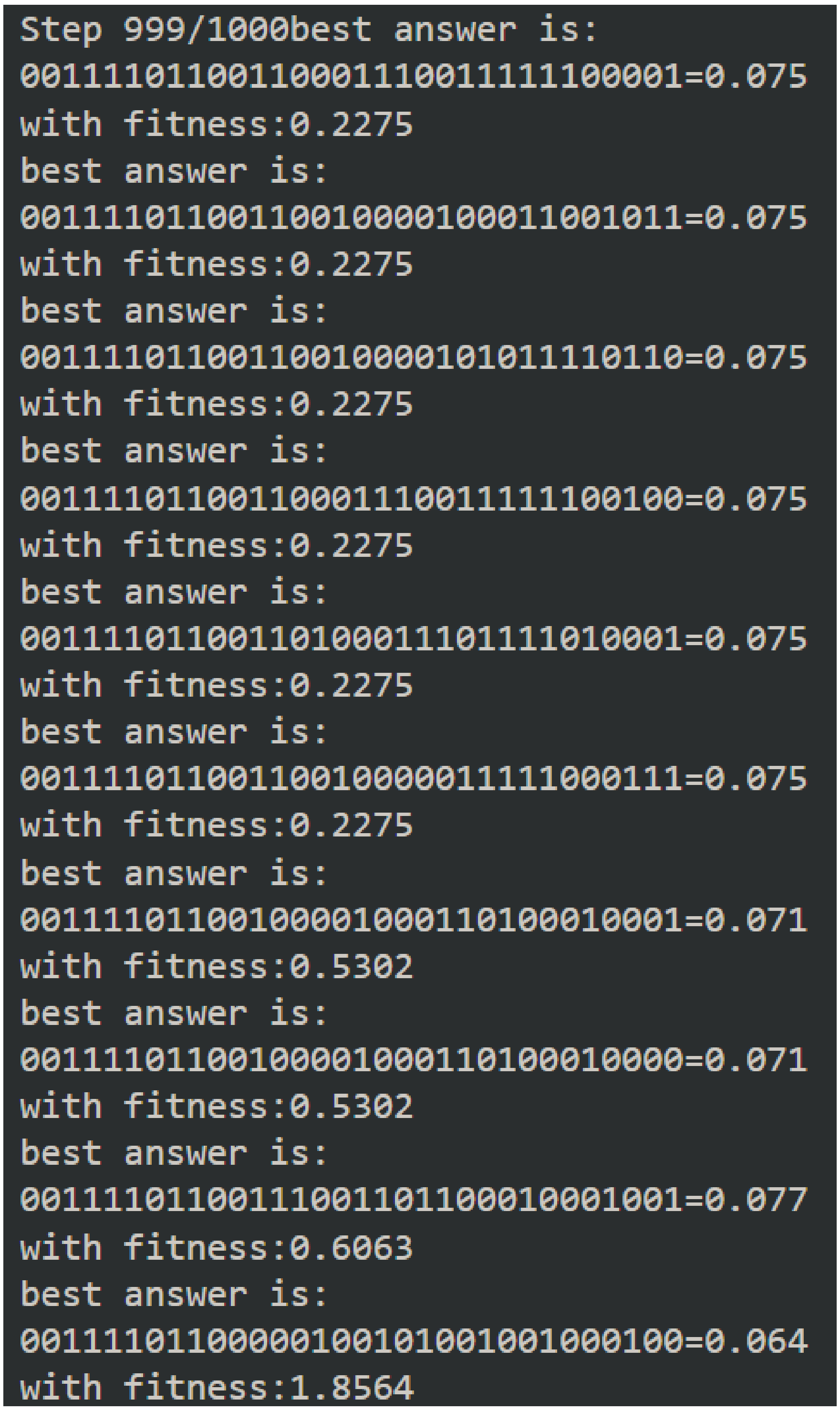
جمعیت اولیه را از ۱۰۰ به ۱۰۰۰۰۰ افزایش میدهیم



```
best answer is:
00111101100110001000110111110110=0.074
with fitness:0.0381
best answer is:
00111101100101110111010011011110=0.074
with fitness:0.0381
best answer is:
00111101100101110111011111110111=0.074
with fitness:0.0381
best answer is:
00111101100101010100000011000101=0.073
with fitness:0.1513
best answer is:
00111101100100111010110100001010=0.072
with fitness:0.3407
best answer is:
00111101100111000110000111011110=0.076
with fitness:0.4169
best answer is:
00111101100100010101101001110000=0.071
with fitness:0.5302
best answer is:
00111101100100001001110010111001=0.071
with fitness:0.5302
best answer is:
00111101100100001000001001011111=0.071
with fitness:0.5302
best answer is:
00111101100111100111011001011110=0.077
with fitness:0.6063
```

شکل ۸:

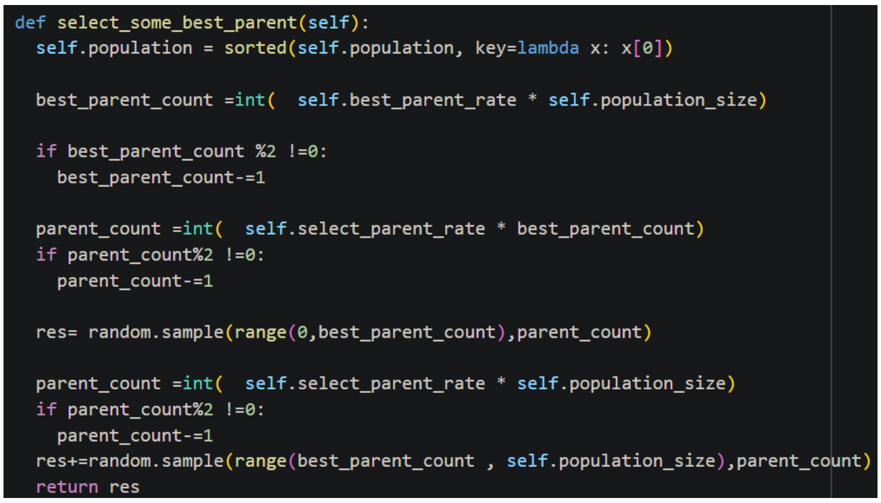
حال مراحل پیمایش را از ۱۰۰ به ۱۰۰۰ افزایش میدهیم



```
Step 999/1000best answer is:
00111101100110001110011111100001=0.075
with fitness:0.2275
best answer is:
00111101100110010000100011001011=0.075
with fitness:0.2275
best answer is:
00111101100110010000101011110110=0.075
with fitness:0.2275
best answer is:
00111101100110001110011111100100=0.075
with fitness:0.2275
best answer is:
00111101100110100011101111010001=0.075
with fitness:0.2275
best answer is:
00111101100110010000011111000111=0.075
with fitness:0.2275
best answer is:
00111101100100001000110100010001=0.071
with fitness:0.5302
best answer is:
00111101100100001000110100010000=0.071
with fitness:0.5302
best answer is:
00111101100111001101100010001001=0.077
with fitness:0.6063
best answer is:
00111101100000100101001001000100=0.064
with fitness:1.8564
```

شکل ۹:

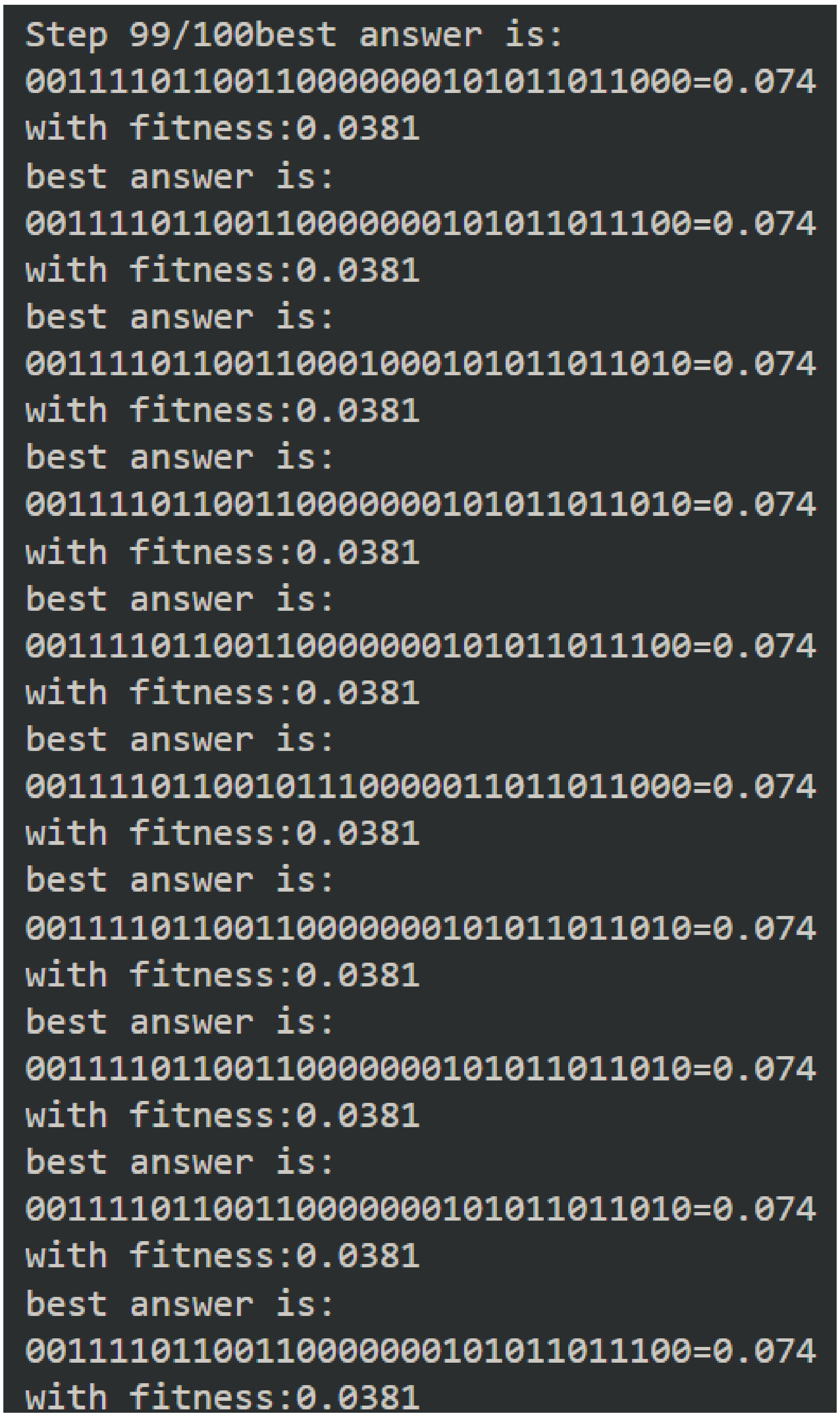
همانطور که میبینیم جواب دوم را حدس نمیزند پس کمی تنظیمات را تغییر داده و به جای انتخاب تصادفی والدین ، با توجه به فیتنس آن را تغییر میدهیم



```python
def select_some_best_parent(self):
  self.population = sorted(self.population, key=lambda x: x[0])

  best_parent_count =int(  self.best_parent_rate * self.population_size)

  if best_parent_count %2 !=0:
    best_parent_count-=1

  parent_count =int(  self.select_parent_rate * best_parent_count)
  if parent_count%2 !=0:
    parent_count-=1

  res= random.sample(range(0,best_parent_count),parent_count)

  parent_count =int(  self.select_parent_rate * self.population_size)
  if parent_count%2 !=0:
    parent_count-=1
  res+=random.sample(range(best_parent_count , self.population_size),parent_count)
  return res
```

شکل ۱۰:

میبینیم با این تغییر در تعداد کمتری از جمعیت و تعداد کل مرحله به جواب همگرا میشود

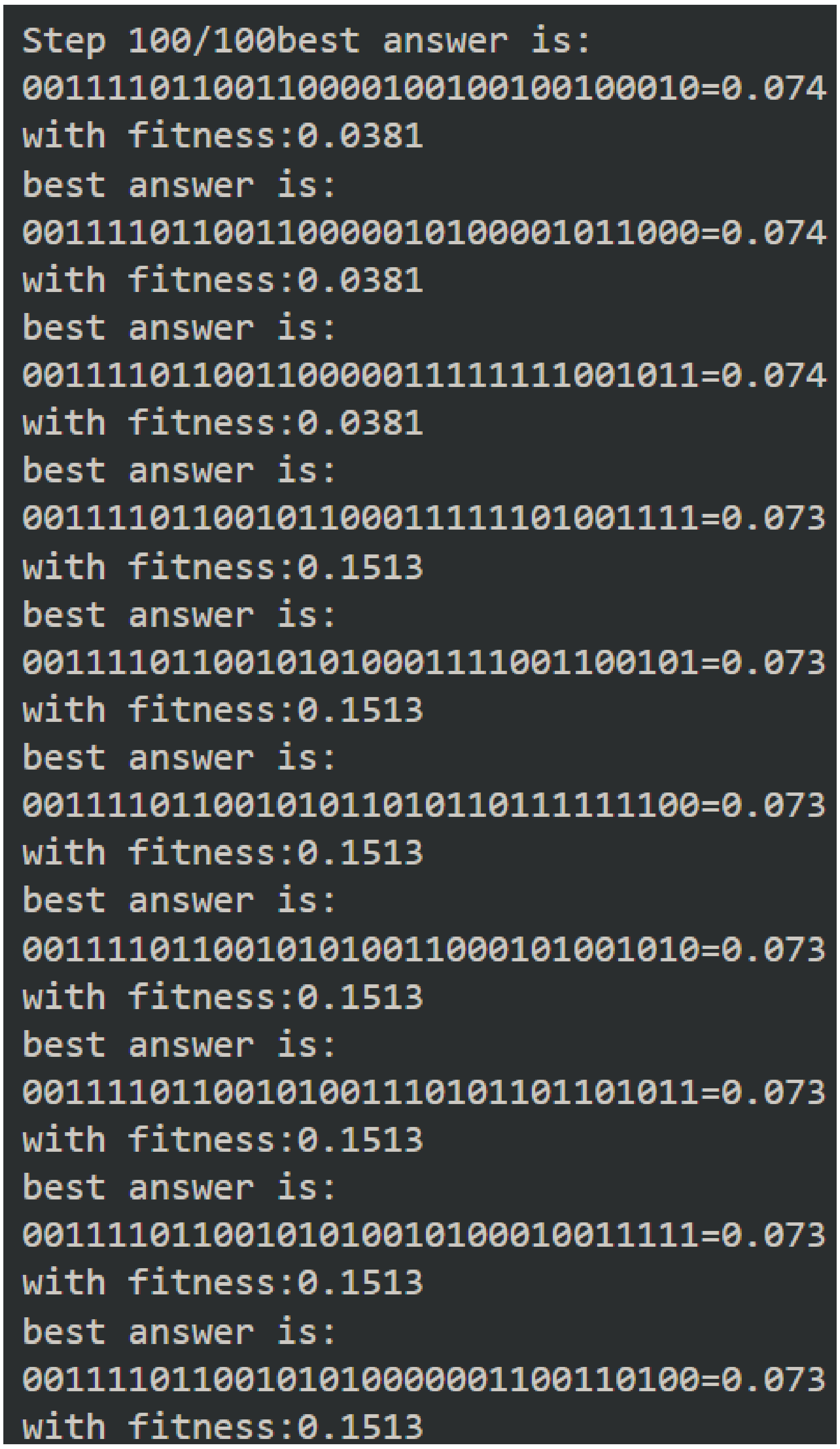


```
Step 99/100best answer is:
00111101100110000000101011011000=0.074
with fitness:0.0381
best answer is:
00111101100110000000101011011100=0.074
with fitness:0.0381
best answer is:
00111101100110001000101011011010=0.074
with fitness:0.0381
best answer is:
00111101100110000000101011011010=0.074
with fitness:0.0381
best answer is:
00111101100110000000101011011100=0.074
with fitness:0.0381
best answer is:
00111101100101110000011011011000=0.074
with fitness:0.0381
best answer is:
00111101100110000000101011011010=0.074
with fitness:0.0381
best answer is:
00111101100110000000101011011010=0.074
with fitness:0.0381
best answer is:
00111101100110000000101011011010=0.074
with fitness:0.0381
best answer is:
00111101100110000000101011011100=0.074
with fitness:0.0381
```

شکل ۱۱:

حال تعداد جمعیت را به ۱۰۰۰۰۰ افزایش میدهیم

ولی باز هم جواب دوم را پیدا نمیکند



```
Step 100/100best answer is:
0011110110011000010010010010010=0.074
with fitness:0.0381
best answer is:
0011110110011000001010000101100=0.074
with fitness:0.0381
best answer is:
0011110110011000001111111001011=0.074
with fitness:0.0381
best answer is:
0011110110010110001111110100111=0.073
with fitness:0.1513
best answer is:
0011110110010101000111100110010=0.073
with fitness:0.1513
best answer is:
0011110110010101101011011111100=0.073
with fitness:0.1513
best answer is:
0011110110010101001100010100101=0.073
with fitness:0.1513
best answer is:
0011110110010100111010110110101=0.073
with fitness:0.1513
best answer is:
0011110110010101001010001001111=0.073
with fitness:0.1513
best answer is:
0011110110010101000000110011010=0.073
with fitness:0.1513
```

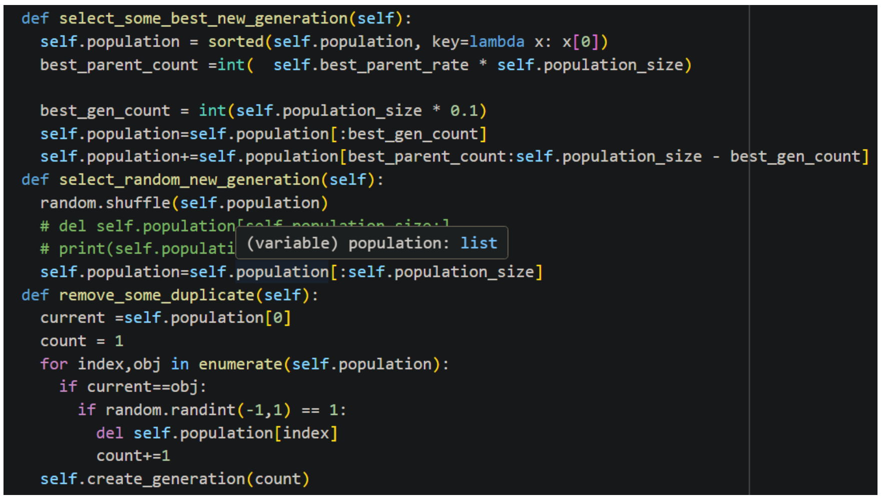
شکل ۱۲:

پس باز هم تنظیمات را تغییر میدهیم

هنگام انتخاب جمعیت نهایی سعی میکنیم کمی از جواب های خوب و کمی از جواب های بد برداریم

در جواب ها مشاهده میکنیم که جواب های برتر دقیقا مثل هم هستند پس جواب های مثل هم را سعی میکنیم آخر هر

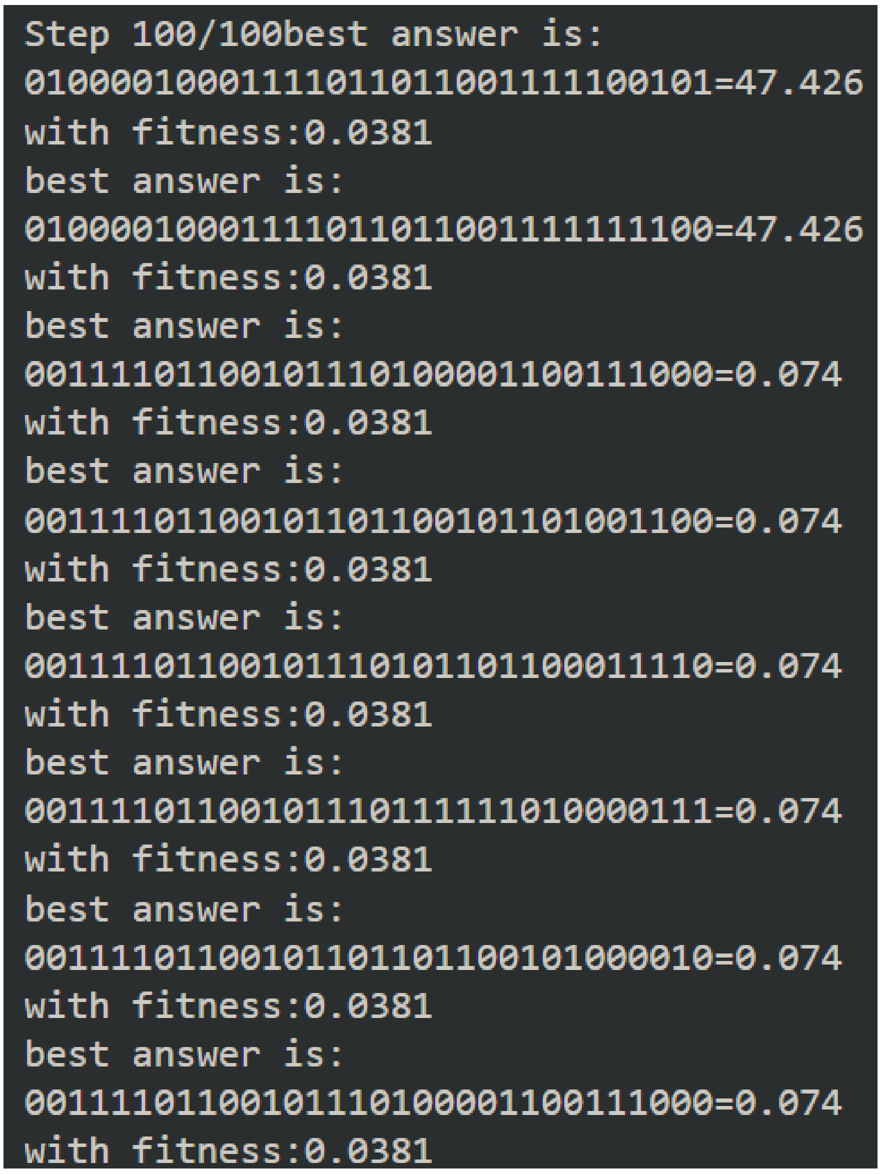
ایپاک با احتمال یک دوم پاک کنیم و جای آن به صورت رندم مثال اضافه کنیم



```python
def select_some_best_new_generation(self):
  self.population = sorted(self.population, key=lambda x: x[0])
  best_parent_count =int(  self.best_parent_rate * self.population_size)

  best_gen_count = int(self.population_size * 0.1)
  self.population=self.population[:best_gen_count]
  self.population+=self.population[best_parent_count:self.population_size - best_gen_count]
def select_random_new_generation(self):
  random.shuffle(self.population)
  # del self.population[self.population_size:]
  # print(self.populati (variable) population: list
  self.population=self.population[:self.population_size]
def remove_some_duplicate(self):
  current =self.population[0]
  count = 1
  for index,obj in enumerate(self.population):
    if current==obj:
      if random.randint(-1,1) == 1:
        del self.population[index]
        count+=1
  self.create_generation(count)
```

شکل ۱۳:

حال با همان پارامتر ها سعی در اجرای مدل میکنیم و مشاهده میکنیم بالاخره جواب دوم را هم پیدا میکند



```
Step 100/100best answer is:
01000010001111011011001111100101=47.426
with fitness:0.0381
best answer is:
01000010001111011011001111111100=47.426
with fitness:0.0381
best answer is:
00111101100101110100001100111000=0.074
with fitness:0.0381
best answer is:
00111101100101101100101101001100=0.074
with fitness:0.0381
best answer is:
00111101100101110101101100011110=0.074
with fitness:0.0381
best answer is:
00111101100101110111111010000111=0.074
with fitness:0.0381
best answer is:
00111101100101101101100101000010=0.074
with fitness:0.0381
best answer is:
00111101100101110100001100111000=0.074
with fitness:0.0381
```

شکل ۱۴: