گزارش کار تمرین عملی GA

امیر حسین باقری ۹۸۱۰۵۶۲۱

A کد اولیه

بخش اصلی الگوریتم

```
def fitness():
  fitness_amount = []
  maximum =  np.inf
  best_string = None
  for dna_string in states:
    temp = 0
    for key, value in constraints.items():
      temp += len(re.findall(str(key), (dna_string)))*value
      if temp > maximum:
        maximum = temp
        temp = 0 if temp < 0 else temp
        best_string = dna_string
    fitness_amount.append(exponential**temp)
  return fitness_amount,maximum,best_string
best_fitness_arr = []
best_string_fitness = None
for i in range(n_iteration):
  fitness_amount,maximum,best_string = fitness()
  best_fitness_arr.append(maximum)
  best_string_fitness = best_string
  #selection
  states = random.choices(states,fitness_amount ,cum_weights=None, k=
    size_of_population)
  #cross over
  for i in range(0,size_of_population, 2):
    first_state,second_state =states[i] ,states[i+1]
    random_index = random.randint(0, lenght_of_DNA _ 1)
    temp1 = first_state[:random_index]+second_state[random_index:]
    temp2 = second_state[:random_index]+first_state[random_index:]
    states[i],states[i+1] = temp1,temp2
  # mutation
  for i in range(size_of_population):
  if random.random() <= 0.01:
    temp = list( states[i])
    temp[random.randint(0, lenght_of_DNA _ 1)] = random.choice(
    nucleotides)
    states[i] = ''.join(temp)
```

توضیحات

برای اینکه در fitness مقادیر منفی بدست می‌آیند بنابراین از exp استفاده کرده ایم.
۳ نکته مهم :
اول آنکه برای اینکه ممکن است مقادیر بدست آمده با تابع توانی از دقت اعداد فلوت در پایتون کمتر باشند و مقدار صفر برای آنها لحاظ شوند برای توان آنها مقدار صفر را در نظر گرفته ایم.
نکته دوم آنکه عبارت بالا را تنها برای مقادیری که از ماکسیمم فیتنس حال حاضر بیشتر است اعمال کرده ایم (هر چند برای تمام نقاط صعودی که خاصیت بالا را دارند یک احتمال را در نظر گرفته ایم اما چون در این بخش نیاز به جواب خیلی خوب نداریم همین کفایت می‌کند.)
برای انتخاب (selection) از تابع random.choices استفاده کرده ایم که یک توالی وزن برای ورودی میگیرد و بر اساس آن انتخاب می‌کند. (نمودار ها در انتها آمده اند.)
نکته سوم آنکه از آنجا که پایه ۲.۵ ممکن است برای برخی مقادیر مقادیر بسیار بزرگ تولید کند در ابتدا یک تخمین از حدود میزنیم بدین صورت که اگر الگویی مقدار امتیاز بیش از ۱ داشت (که می تواند موجب امتیاز زیاد شود نمارا به ۱.۳ تغیر می‌دهیم.)

```
if can_be_fitness_more_than_1:
  exponential = 1.3

```

B کد بهبود یافته

در اینجا تنها به بخش های تغیر یافته و توضیحات آنها می پردازیم.

بخش های تغیر یافته

اندازه جمعیت

اندازه جمعیت که همان اندازه پدر هاست دوبرابر کرده ایم.

قسمت fitness

```
fitness_amount = []
maximum = _ np.inf
best_string = None
counter = 0
  for dna_string in states:
  temp = 0
  for key, value in constraints.items():
    temp += len(re.findall(str(key), (dna_string)))*value
    if temp > maximum:
      maximum = temp
      best_string = dna_string
      temp = counter if temp <= 0 else temp
      counter+=1
  fit_amount = ((temp+1 if temp >=0 else  ( temp)** _ 1)*(exponential**
    temp))
  fitness_amount.append(fit_amount)
return fitness_amount,maximum,best_string
```

در این قسمت دو تغیر داده ایم:
اول آنکه تابع فیتنس را طوری تغیر داده ایم که به مقادیر دارای بیشتر بیشتر از مقدار اولیه که نسبت می‌داده ایم نسبت دهد یعنی احتمال انتخاب آنها را از مقداری که به وسیله fitness معمولی تعیین میشد فراتر بردیم.

$$((temp + 1\ if\ temp\ >= 0\ else\ -temp) * * -1) * (exponential * *temp))$$

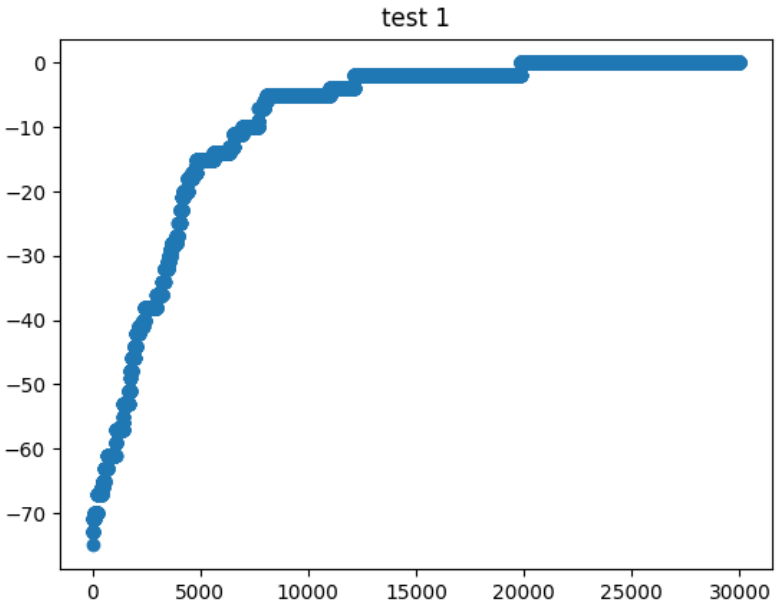
دقت کنید که در رابطه بالا مقادیر منفی هر چی منفی تر باشند کسر کمتر می‌شود و مقادیر مثبت هر چه مثبت تر شوند کسر بزرگتر
نکته دوم آنکه در قسمت A به مقادیری که در دقت اعشاری ممکن بود صفر تعبیر شوند یک مقدار وزن داده ایم ولی در این جا یک شمارنده قرار داده ایم و وزن های بیشتری به فیتنس های بهتر نسبت می‌دهیم.

قسمت mutation

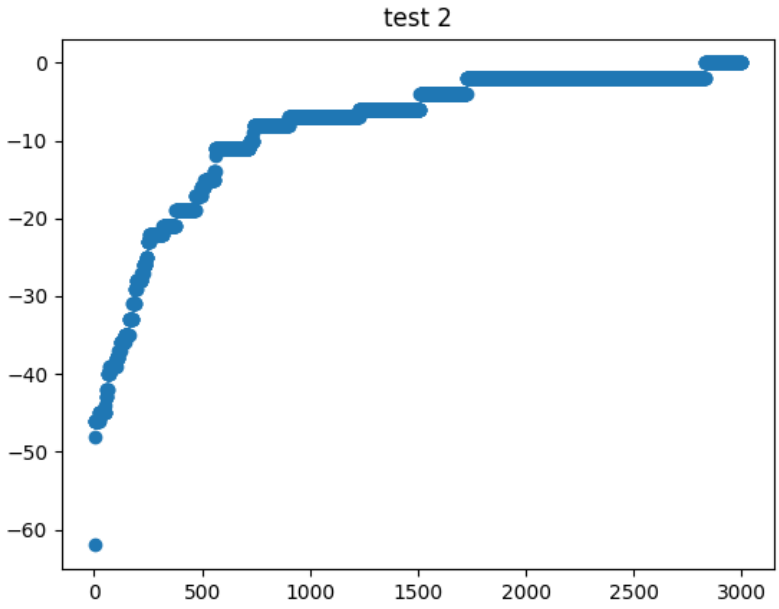
```
for i in range(size_of_population):
  if random.random() <= 0.2:
    temp = list( states[i])
    temp[random.randint(0, lenght_of_DNA _ 1)] = random.choice(
    nucleotides)
    states[i] = ''.join(temp)

```

احتمال را به ۲۰ درصد افزایش داده ایم.
بنابراین **پارامتر های fitness و اندازه جمعیت و mutation را تغیر داده ایم.**
دقت کنید که در هردو قسمت A ، B تغیرات بخش fitness در selection خود را نشان می‌دهند
در زیپ ارسالی پوشه extra کد هر دو بخش به همراه عکس نمودار ها و همچنین فایل txt شامل رشته و مقدار زمان صرف شده قرار دارد.

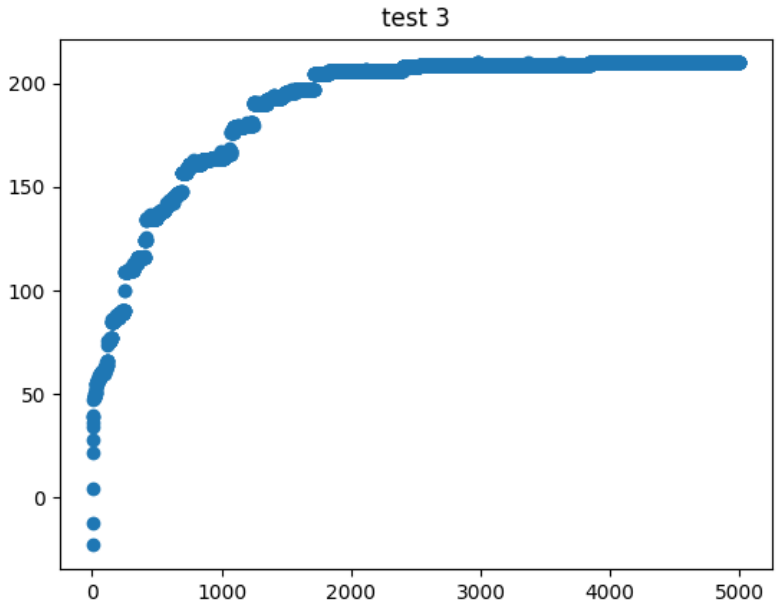
نمودارات مربوط به بخش A



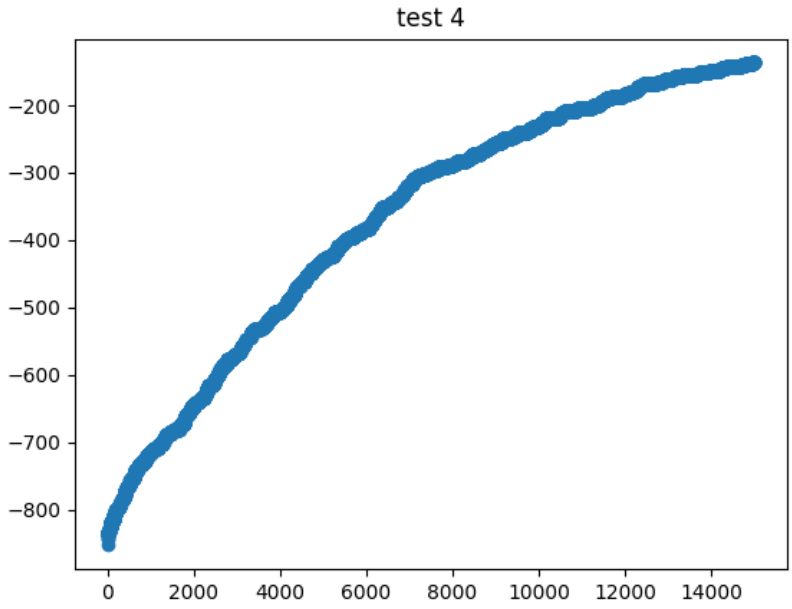
A۱



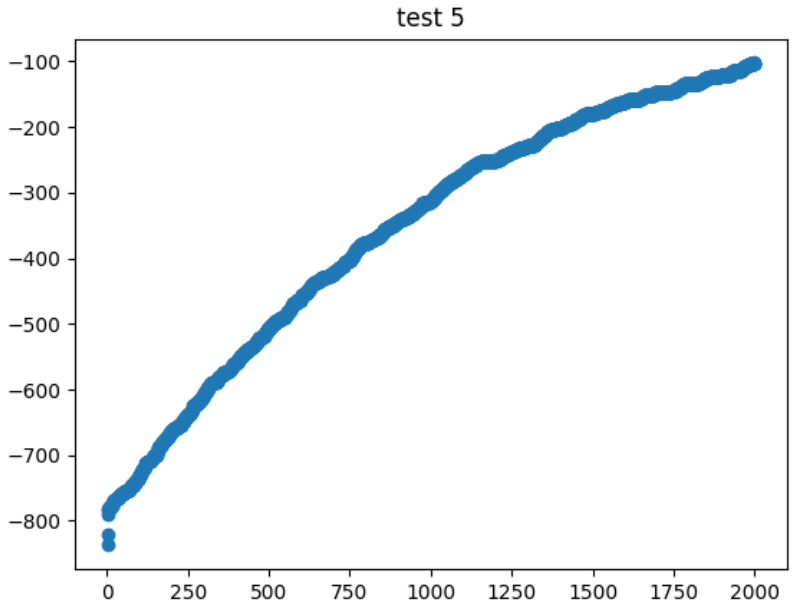
A۲



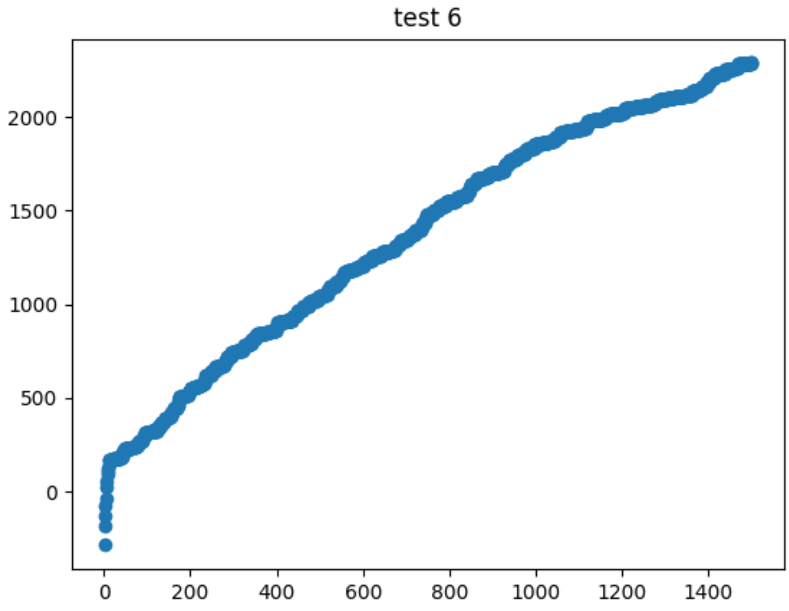
A۳


test 4
−200
−300
−400
−500
−600
−700
−800
0
2000
4000
6000
8000
10000
12000
14000

A۴

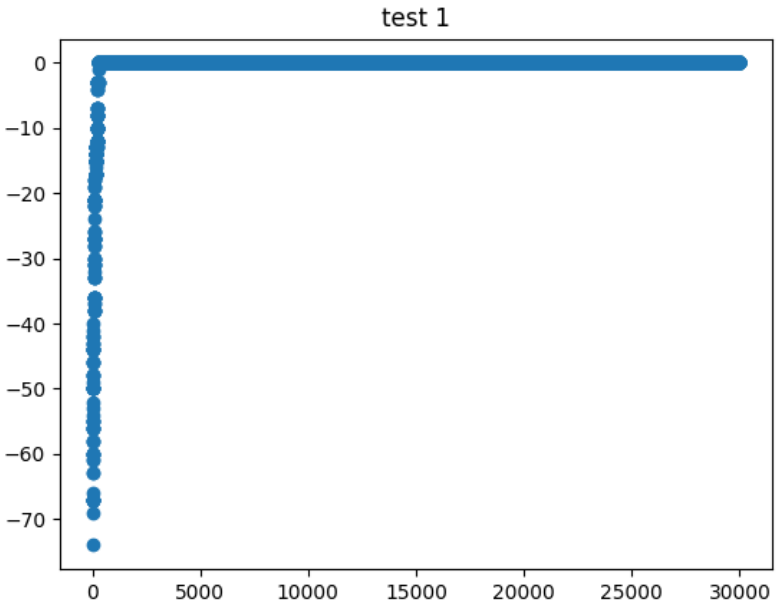

test 5
−100
−200
−300
−400
−500
−600
−700
−800
0
250
500
750
1000
1250
1500
1750
2000

A۵

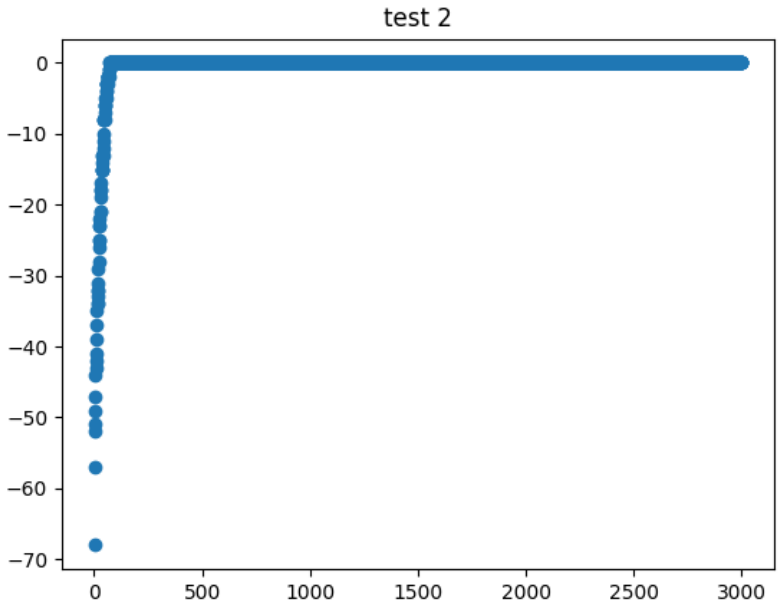


A۶

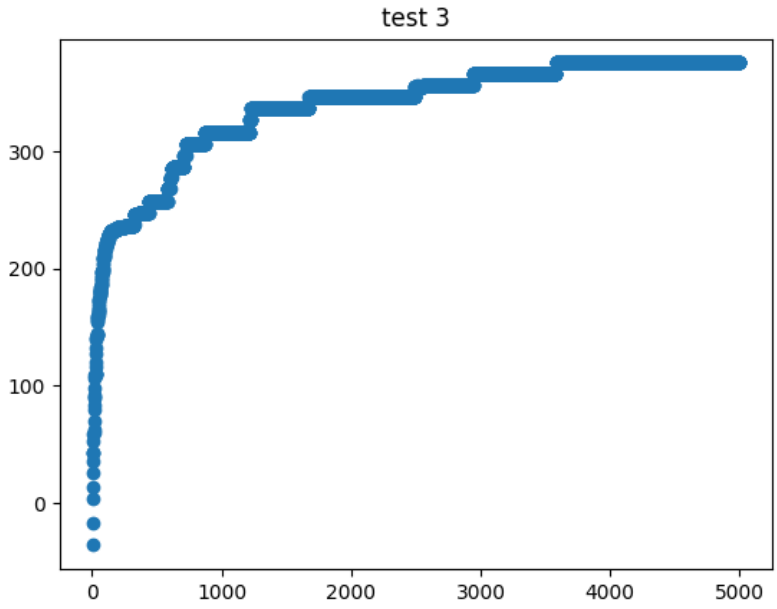
نمودارات مربوط به بخش B



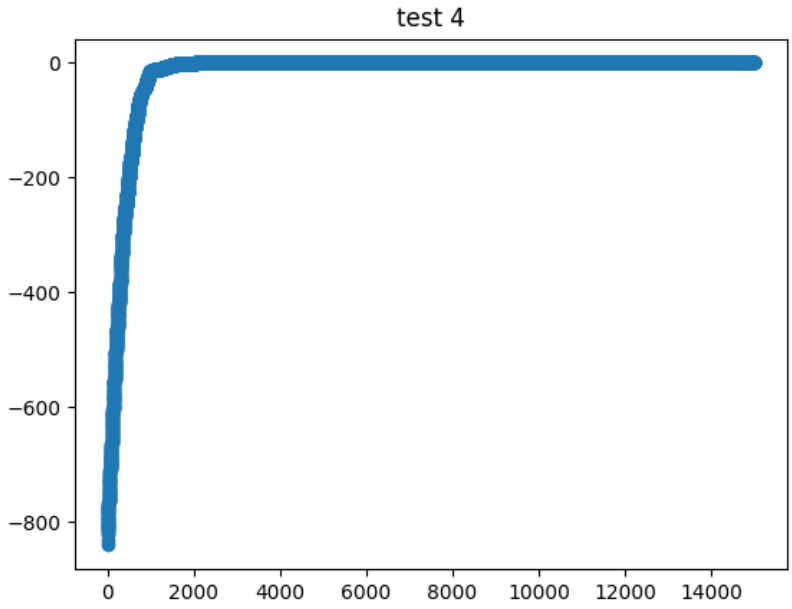
B۱



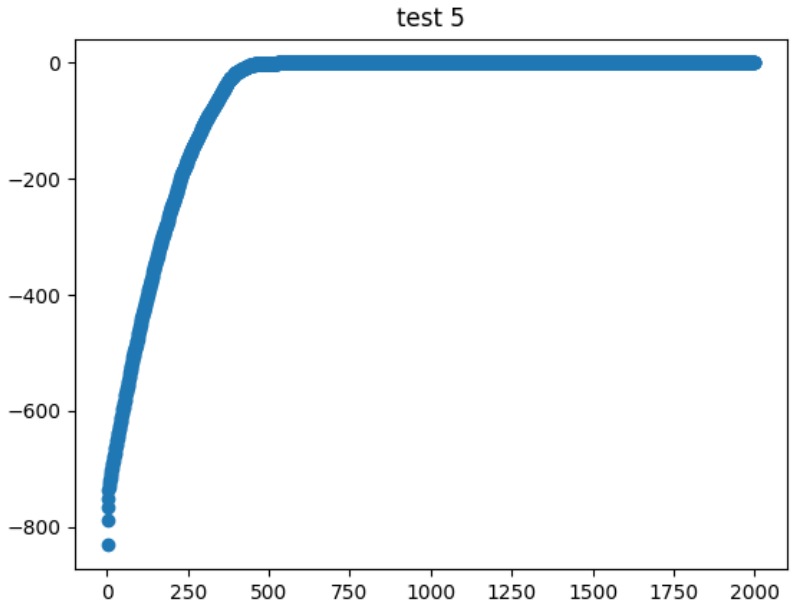
B۲



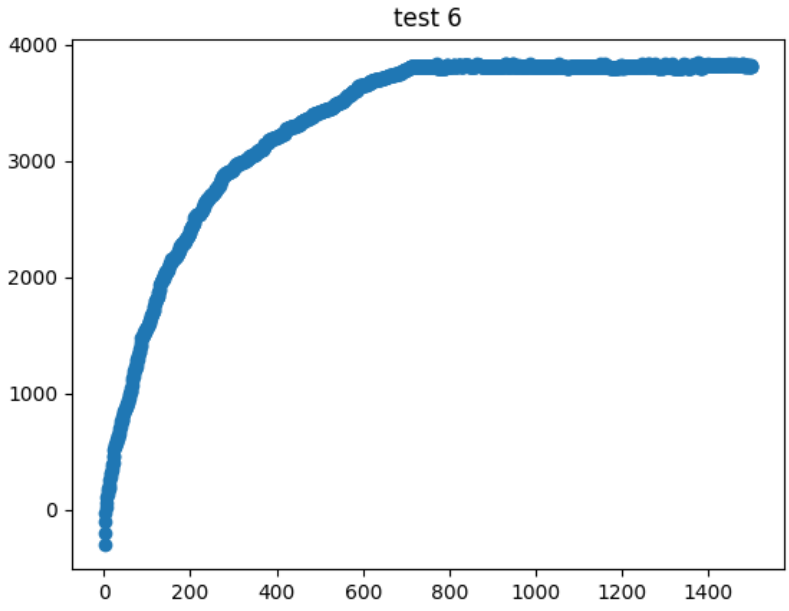
B۳


test 4
0
−200
−400
−600
−800
0
2000
4000
6000
8000
10000
12000
14000

B۴



B۵



B۶