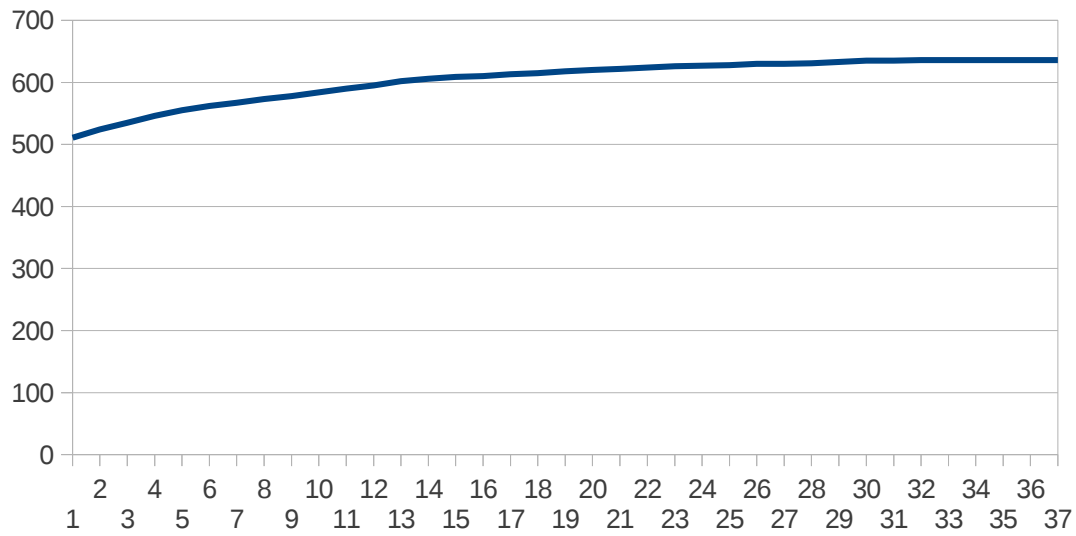


1. Diagram: Average genes sum in hromosomes by ineration(number of generation)

In x-line we have current iteration(number of generation). In y-line we have value of Average sum



of genes in hromosomes

Diagram: max hromosomes sum by iteration(number of generation)

In x-line we have current iteration(number of generation). In y-line we have value of max sum of genes in hromosomes

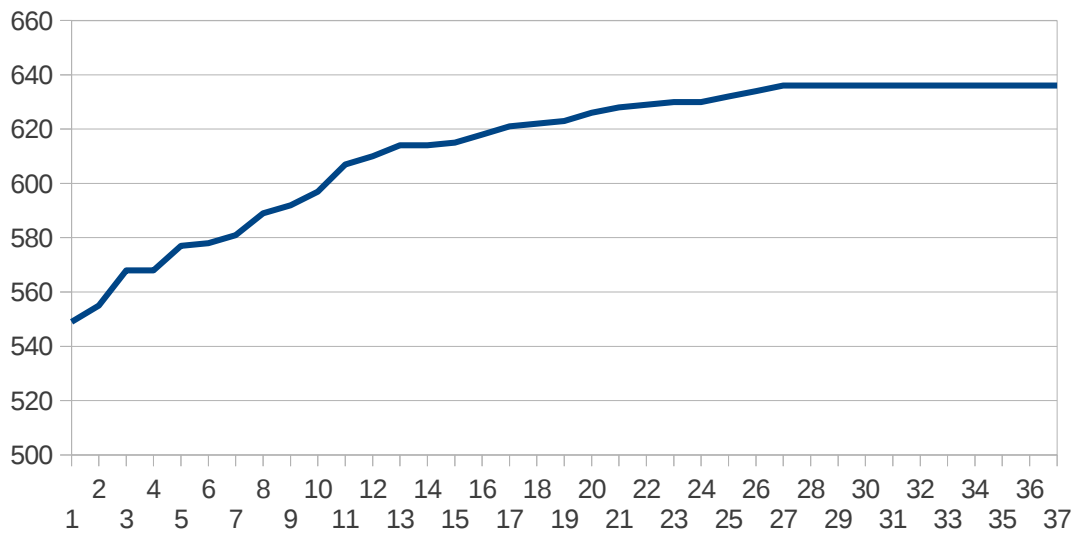


Table: result from program:

ITERATION	AVERAGE	MAX
1	511	549
2	524	555
3	535	568
4	546	568
5	555	577
6	562	578
7	567	581
8	573	589
9	578	592
10	584	597
11	590	607
12	595	610
13	602	614
14	606	614
15	609	615
16	610	618
17	613	621
18	615	622
19	618	623
20	620	626
21	622	628
22	624	629
23	626	630
24	627	630
25	628	632
26	630	634
27	630	636
28	631	636
29	633	636
30	635	636
31	635	636
32	636	636
33	636	636
34	636	636
35	636	636
36	636	636
37	636	636

Source code:

```
import random
def get_average_value(generation):
    av_list = []
    for i in generation:
        sum = 0
        for j in i:
            sum += j
        av_list.append(sum)
    average = reduce(lambda x,y: x+y,av_list)/len(av_list)
    max_value = max(av_list)
    return (average,max_value)

def main():
    generation = []
    for i in xrange(0,100):
        hromosome = []
        for j in xrange(0,1000):
            hromosome.append(random.getrandbits(1))
        generation.append(hromosome)

    max_value = get_average_value(generation)[1]
    print(max_value)
    count = 0
    while count < 10:
        generation.sort(key=lambda x: reduce(lambda y,z: y+z,x), reverse = True)
        fathers = generation[0:50]
        new_generation = []
        for i in xrange(0,50):
            mother = random.randrange(0,50)
            father = random.randrange(0,50)
            rand_index = random.randrange(0,1000)
            first = fathers[mother][0:rand_index]
            first.extend(fathers[father][rand_index:1000])
            second = fathers[father][0:rand_index]
            second.extend(fathers[mother][rand_index:1000])
            new_generation.append(first)
            new_generation.append(second)
        new_generation.extend(fathers)
        new_max = get_average_value(new_generation)
        print(new_max)
        if new_max[1] > max_value:
            max_value = new_max[1]
            count = 0
        else:
            count += 1
        generation = new_generation

if __name__ == '__main__':
    main()
```