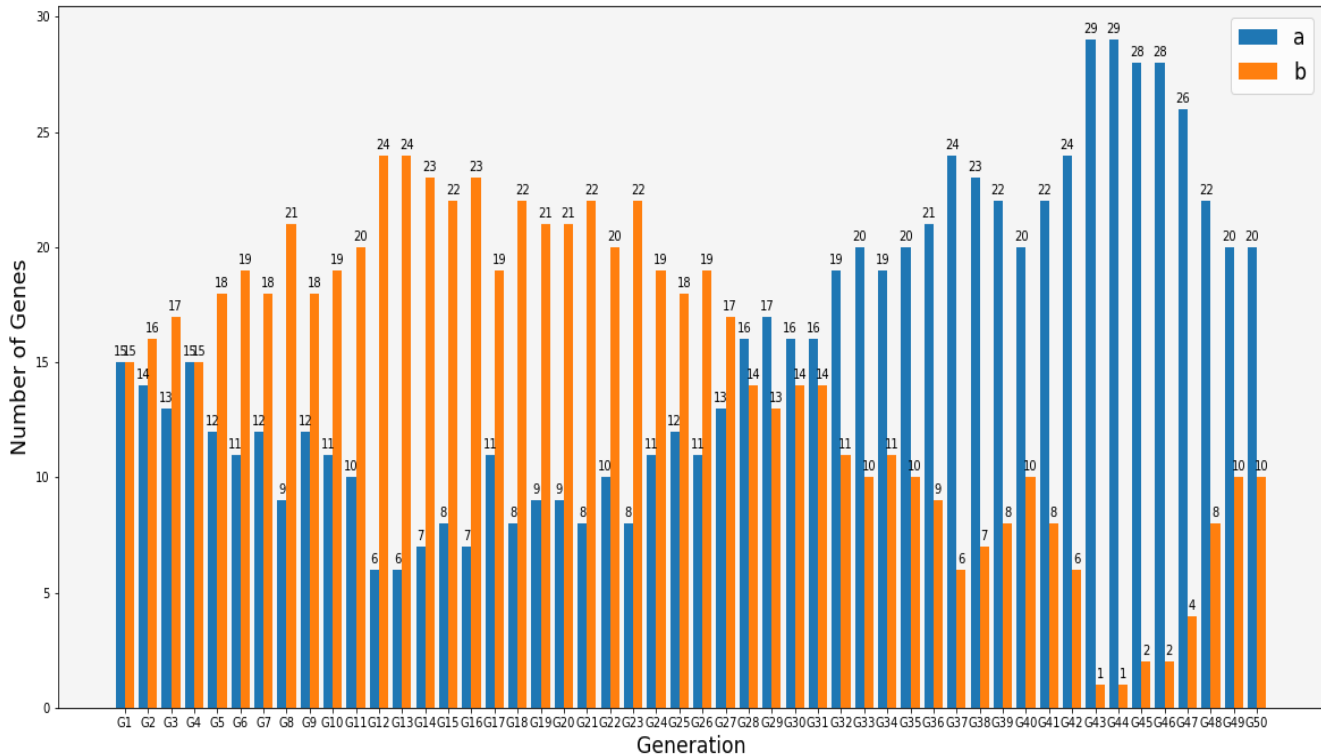
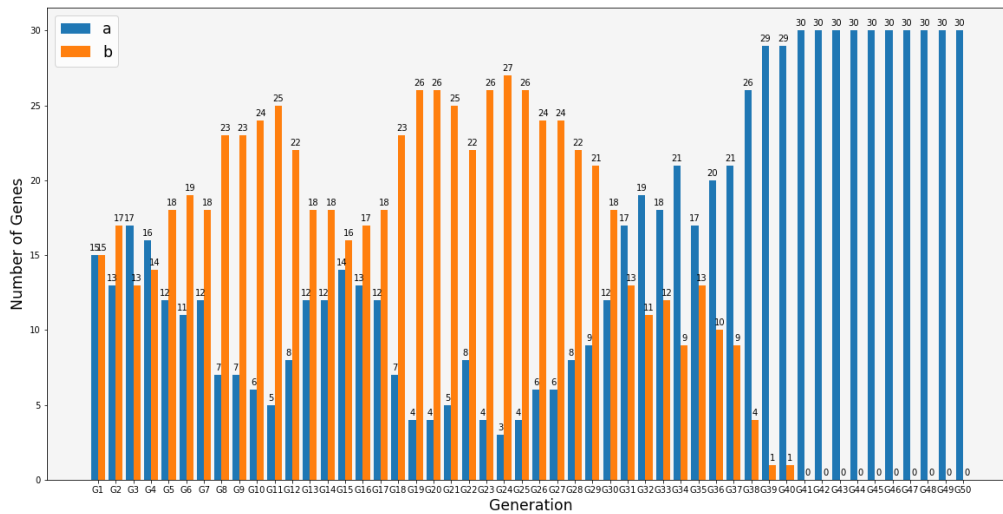
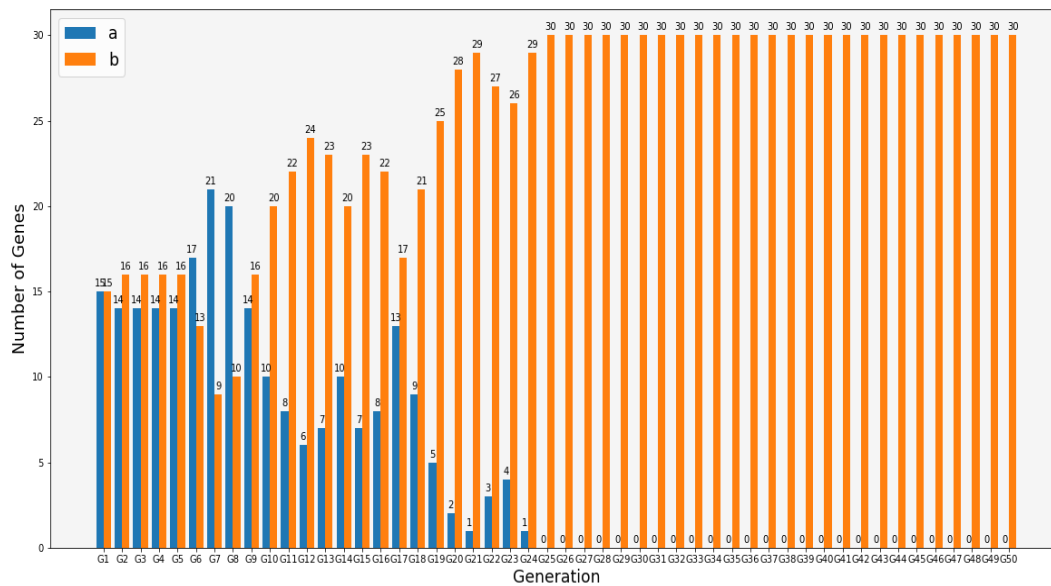
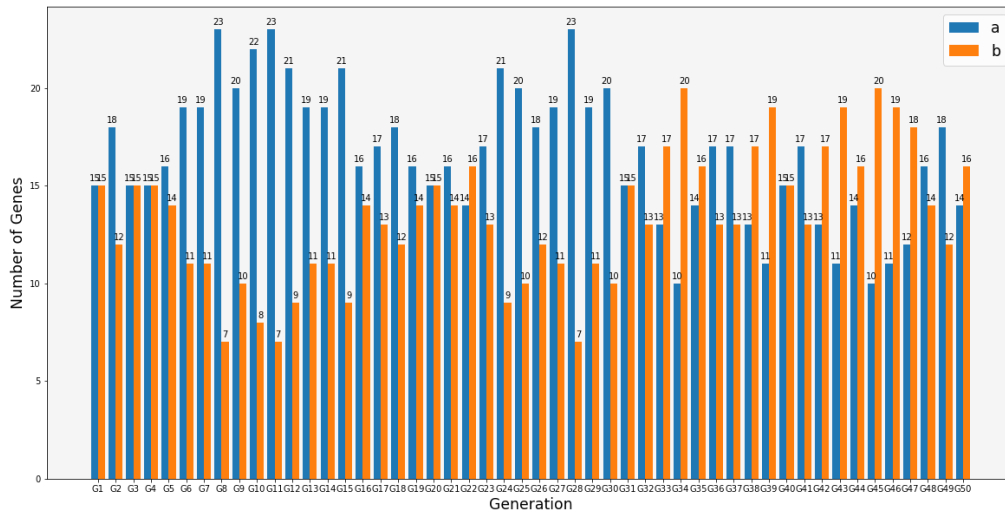
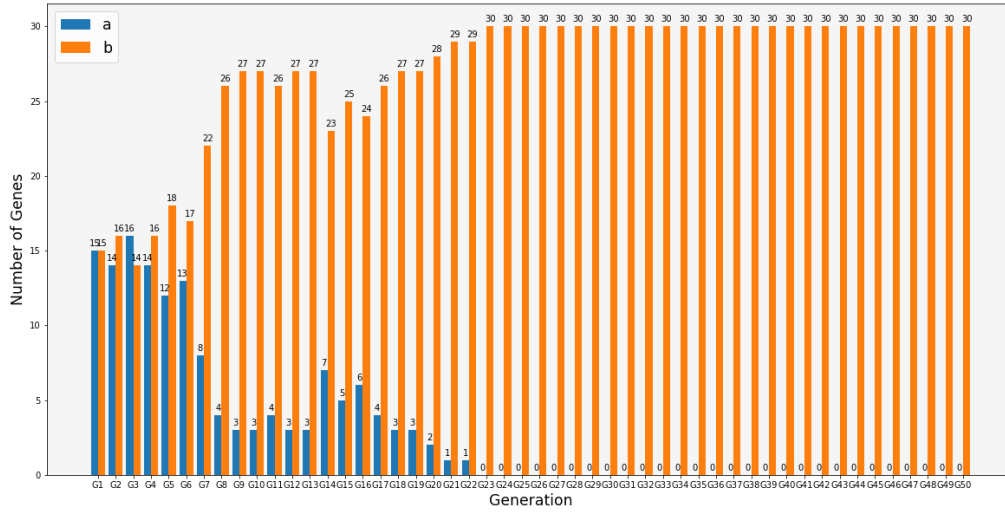


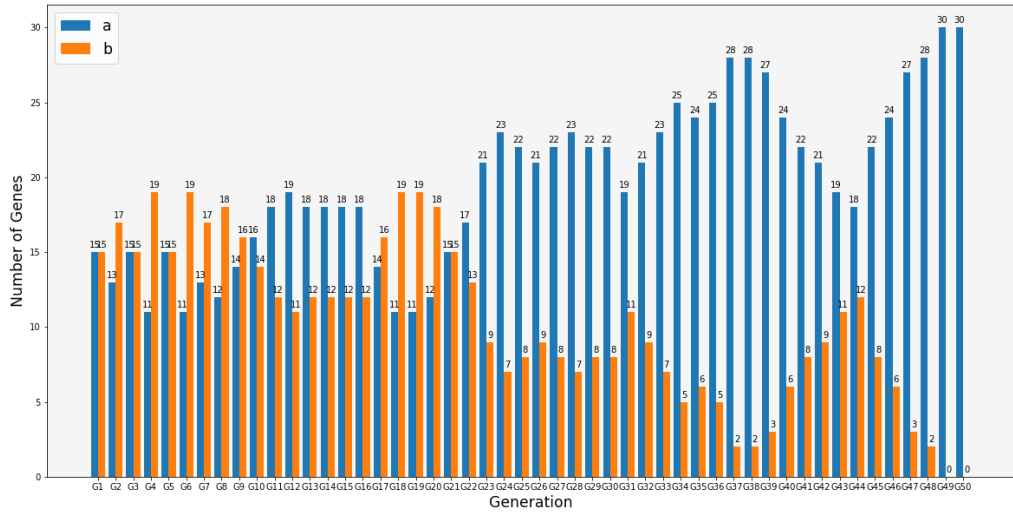
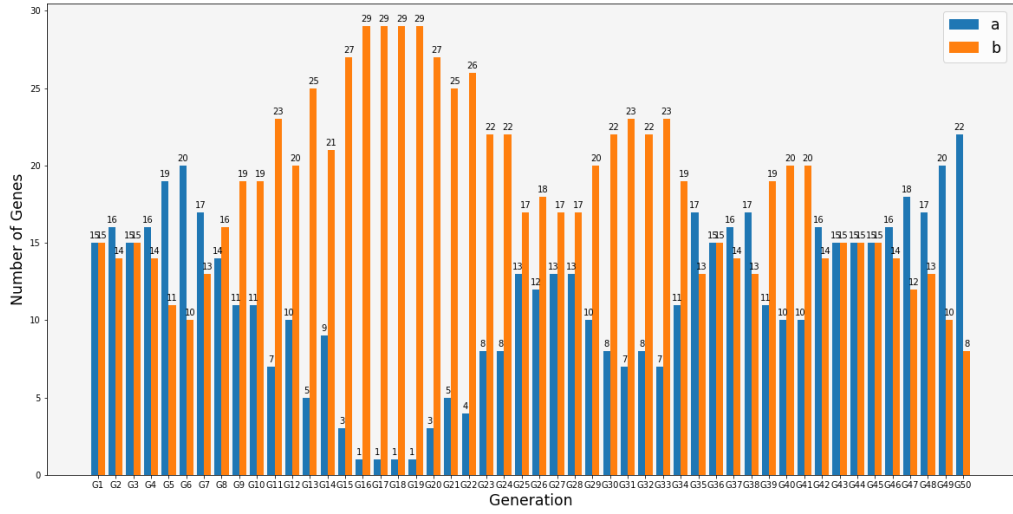
در ژنتیک جامعه، تکامل به صورت تغییر در فرکانس **allele**‌های یک ژن در جامعه در طی نسل‌ها تعریف می‌شود. این تغییر می‌تواند به واسطه انتخاب طبیعی و یا مکانیزم‌های دیگری صورت بگیرد و ممکن است جامعه را برای محیطش مناسب‌تر و یا نامناسب‌تر بکند. در ژنتیک مفهومی وجود دارد به نام "Genetic Drift" و به معنای تغییر در فرکانس **allele**‌های یک ژن از نسلی به نسل دیگر به واسطه شانس است. به بیان دقیق‌تر به معنای تغییر به واسطه خطای نمونه‌برداری از مجموعه ژن موجود برای نسل بعدی است. این پدیده در تمام جوامع متناهی اتفاق می‌افتد ولی تاثیر آن در جوامع کوچک بیشتر دیده می‌شود. Genetic drift می‌تواند منجر به حذف کامل یک **allele** (احتمالاً **allele**‌های سودمند) از جامعه و یا به ۱۰۰٪ رسیدن سایر **allele**‌ها شود که این پدیده **fixation** نام دارد.

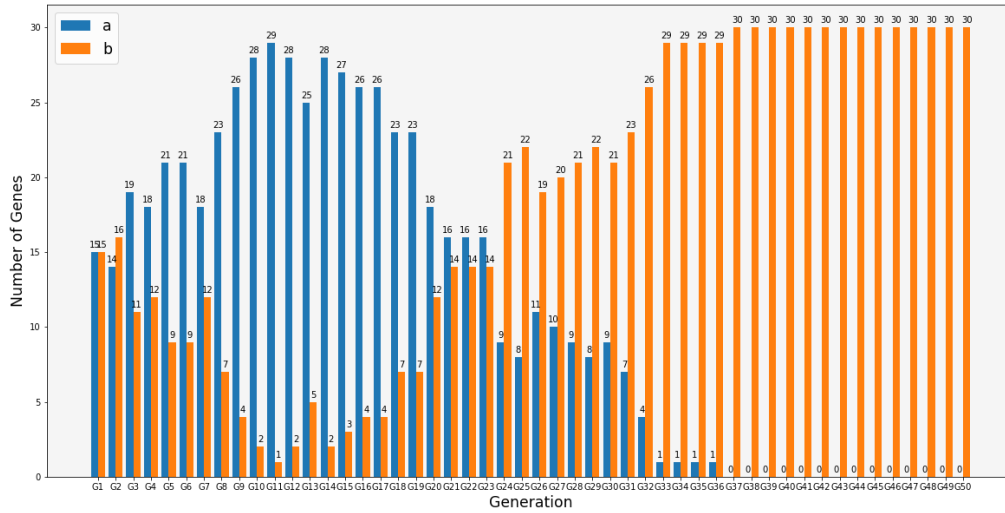
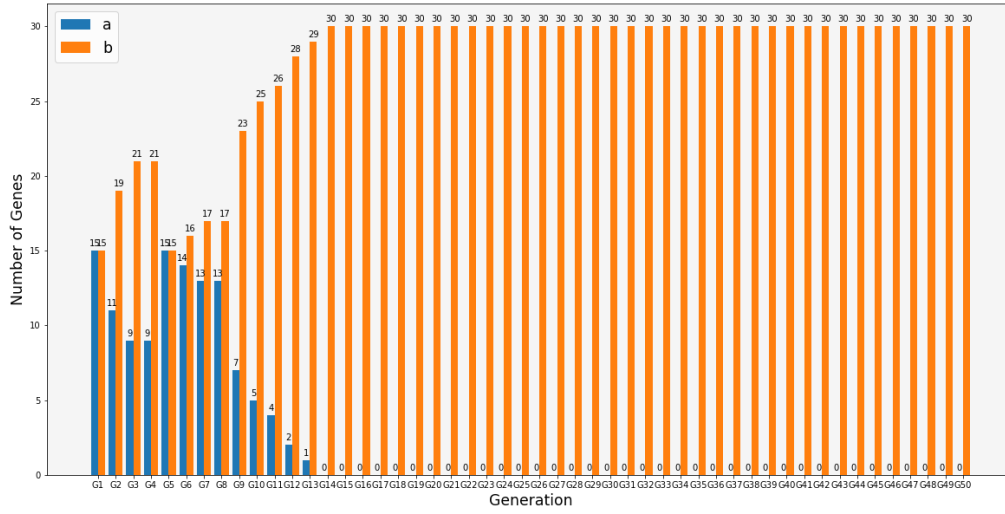
در این تکلیف سعی شده‌است که **genetic drift** را در یک جامعه با ۳۰ عضو شبیه سازی کنیم. به این صورت عمل کردیم که در ابتدا ۱۵ عضو جامعه دارای **allele** از نوع **a** و ۱۵ عضو دارای **allele** از نوع **b** هستند و سپس به صورت تصادفی و با جایگذاری ۳۰ عضو از این مجموعه ژن انتخاب و به نسل بعد منتقل کردیم. این عمل را برای ۵۰ نسل تکرار کردیم و کل آزمایش را ۱۰ بار تکرار کردیم و نتایج ۱۰ تکرار در نمودارهای زیر آورده شده‌است. هر دو میله عمودی آبی و نارنجی در کنار هم بیانگر تعداد **allele**‌های **a** و **b** در یک نسل هستند و بر روی هر میله تعداد آن **allele** آمده‌است.

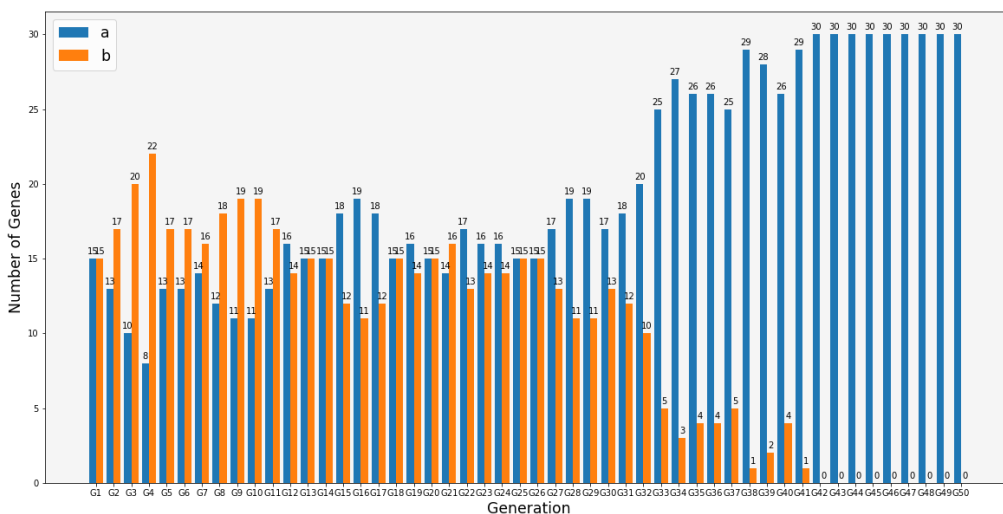












در تعداد قابل توجهی از این آزمایش‌ها دیده می‌شود که در طی زمان یکی از این allele‌ها به طور کامل از بین می‌رود و در تعدادی نیز هر دو allele تا نسل ۵۰ام دیده می‌شوند.

کد برنامه در ادامه آمده‌است.

```
import random
import matplotlib.pyplot as plt

def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its
    height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}' .format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3), # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')

for f in range(10):
    population = [0, 1]*15
    populations = [population]
    generation = [0]
```

```

for i in range(1, 50):
    generation.append(i)
    populations.append(random.choices(populations[-1], k=30))
a_gene_counts = [population.count(0) for population in populations]
b_gene_counts = [population.count(1) for population in populations]

width = 0.8
x_label = [f'G{i}' for i in range(1, len(a_gene_counts)+1)]
x = list(range(1, 2*len(a_gene_counts)+1, 2))
a_x = list(map(lambda x: x - width/2, x))
b_x = list(map(lambda x: x + width/2, x))
fig, ax = plt.subplots(figsize=(20, 10))
rects_1 = ax.bar(a_x, a_gene_counts, width, label='a')
rects_2 = ax.bar(b_x, b_gene_counts, width, label='b')
autolabel(rects_1)
autolabel(rects_2)
ax.set_xticks(x)
ax.set_xticklabels(x_label)
ax.legend(fontsize='xx-large')
ax.set_facecolor("#f5f5f5")
ax.set_ylabel("Number of Genes", fontsize='xx-large')
ax.set_xlabel("Generation", fontsize='xx-large')

```