例 6.9 Python 程序代码

本程序主要针对 M/M/m 模型，例题数据可做适当修改。

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns  # 用于合并绘图

# 定义函数
class P():  # 建立要接受服务的人对象
    def __init__(self, Num, A, WK, WT, ET, FS, ST, R):
        self.Num = Num  # 编号
        self.A = A  # 到达时间 时间点
        self.WT = WT  # 等待时间 时间长度
        self.WK = WK  # “工作”时间 时间长度
        self.ET = ET  # 服务台无人空白时间 时间长度
        self.FS = FS  # 完成工作时间 时间点
        self.ST = ST  # 开始工作时间 时间点
        self.R = R  # 剩余时间 时间长度


def toilet_which(service):  # 返回服务台队列中等待时间最短的索引
    lt = []
    for i in service:
        lt.append(i.R)
    return lt.index(min(lt))


def service_minus_test(service, m):  #
    h = []
    for i in range(M):
        if service[i] != None:
            h.append(service[i].R - m)
    if min(h) <= 0:
        return True
    else:
        return False


def service_None(service):
    return [i for i, x in enumerate(service) if x == None]


def service_0(service, m):
    return [i for i, x in enumerate(service) if x != None and x.R <= m]
```

```
def Nature_minus(x, y):  # 自然数集中的减法
    if x > y:
        return x - y
    else:
        return 0



#
N = 200  # item 数量
M = 12  # 系统中处理 item 的个数
K = 30

np.random.seed(2333)

WK = np.random.uniform(10, 5, size=N)  # 工作时间随机生成
A = np.random.uniform(0, K, size=N)  # item 到达时间随机生成
A.sort()

y = []
for i in range(0, N - 1):
    y.append(A[i + 1] - A[i])
y = np.array(y)

Queue = [P(i, A[i], WK[i], 0, 0, 0, 0, 0) for i in range(N)]  # 初始化 Queue

# 对 toilet 初始化
Queue[0].ST = Queue[0].A
Queue[0].WT = 0
Queue[0].ET = Queue[0].A
Queue[0].R = Queue[0].WK

service = [Queue[0]] + [None for i in range(M - 1)]
lt = []  # 等待的队伍
for k in range(1, N):  # item 陆续进入处理器与等待队伍
    m = y[k - 1]
    if service_minus_test(service, m):

        # print("service",service,end="")
        s = list(set((service_None(service) + service_0(service, m))))
        # print("这是关于 A 的 Num",Queue[k].Num,"时刻为",Queue[k].A)

        if len(lt) == 0:
            print("1,1", Queue[k].Num)
```

```
        v = s[0]
        for i in range(M):
            if i == v and service[v] == None:
                Queue[k].ET = Queue[k].A
                Queue[k].ST = Queue[k].A
                service[v] = Queue[k]
            if i == v and service[v] != None:
                Queue[k].ET = m - service[v].R
                Queue[k].ST = Queue[k].A
                Queue[k].R = Queue[k].WK
                service[v].FS = service[v].ST + service[v].WK
                service[v].R = 0
                service[v] = Queue[k]
            if i != v:
                if service[i] != None:
                    service[i].R = Nature_minus(service[i].R, m)
                    if service[i].R == 0:
                        service[i].FS = service[i].ST + service[i].WK
                        service[i] = None
    else:
        lt.append(Queue[k])
        # print("1,0apend",Queue[k].Num)
        for i in range(M):
            if i in s:
                if len(lt) > 1:
                    service[i].FS = service[i].ST + service[i].WK
                    # print("前 lt",lt[0].Num)
                    r = lt.pop(0)
                    # print("s",s)
                    # print("后 lt",lt[0])

                    # print("第{}号服务台的{}完成".format(i,toilet[i].Num))
                    # print("取出",r.Num,"进入{}服务台".format(i))
                    r.ST = service[i].FS
                    r.ET = 0
                    r.R = r.WK - (m - service[i].R)
                    service[i].R = 0
                    service[i] = r
                    # if k == 10:
                    # print("ST",toilet[i].ST,toilet[i].Num,k)
                if len(lt) == 1:
                    service[i].FS = service[i].ST + service[i].WK
                    service[i].R = 0
                    e = lt.pop(0)
```

```
                        e.ST = e.A
                        e.R = e.WK
                        service[i] = e
                    if len(lt) == 0:
                        service[i].FS = service[i].ST +service[i].WK
                        service[i].R = 0
                        service[i] = None


                else:
                    service[i].R = Nature_minus(service[i].R, m)


    else:
        B = None in service
        # if k ==2 :
        # L = toilet[2]
        # K = B
        # print(toilet[2])
        if B:
            # print("0,1",Queue[k].Num)
            v = service_None(service)
            for i in range(M):
                if i == v[0]:
                    Queue[k].ST = Queue[k].A
                    Queue[k].ET = 0
                    Queue[k].R = Queue[k].WK
                    service[i] = Queue[k]
                    # if k==2:
                    # print("asjdhaskjdhasd")
                if i not in v:
                    service[i].R = Nature_minus(service[i].R, m)

        else:
            # print("0,0",Queue[k].Num)
            for i in range(M):
                service[i].R = Nature_minus(service[i].R, m)
            lt.append(Queue[k])
            # print("append",Queue[k].Num)
    # for i in range(M):
    #   if toilet[i] != None:
    # print("Num,",toilet[i].Num,"R",toilet[i].R)
    # print("m",m)
while len(lt) != 0:  # item 已经到达，进入等待队伍
```

```
    # print("lt 的长度",len(lt))
    v = toilet_which(service)
    x = service[v].R
    for i in range(M):
        if i == v:
            # print(toilet[v].Num,"在",v,"号服务台服务完",end="")
            service[v].FS = service[v].ST + service[v].WK
            service[v].R = 0
            r = lt.pop(0)
            # print(r.Num,"进入{}号服务台".format(v))
            r.ST = service[v].FS
            r.ET = 0
            r.R = r.WK
            service[v] = r
        else:
            service[i].R = Nature_minus(service[i].R, x)
for i in range(M):  # 处理器中剩下的工作
    service[i].FS = service[i].ST + service[i].WK
    service[i].R = 0
    # print(toilet[i].Num,"在",i,"号服务台服务完",end="")

A = []
ST = []
WK = []
FS = []
WT = []
for i in range(N):
    A.append(Queue[i].A)
    ST.append(Queue[i].ST)
    WK.append(Queue[i].WK)
    FS.append(Queue[i].FS)
    WT.append(Queue[i].WT)
A = np.array(A)
ST = np.array(ST)
WK = np.array(WK)
FS = np.array(FS)
WT = np.array(WT)
WT = ST - A

sns.set(style="ticks", context="notebook")
fig = plt.figure(figsize=(8, 6))
arrivingtime, = plt.plot(A, label="arrivingtime")
startingtime, = plt.plot(ST, label='startingtime')
workingtime, = plt.plot(WK, label='workingtime')
```

```
finishtime, = plt.plot(FS, label='finishtime')
waitingtime, = plt.plot(WT, label='waitingtime')
plt.title(("Queuing problem random simulation experiment about {} people to {}
toilets".format(N, M)).title())
plt.xlabel("Arriving Time(min)")
plt.ylabel("Total Time(min)")

plt.legend(handles=[arrivingtime, startingtime, workingtime, finishtime,
waitingtime],
           loc='upper left')

plt.show()
print("{}个服务台{}分钟，对于{}人，平均每人等待{}分钟".format(M, K, N,
np.mean(WT)))
```