



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«МИРЭА – Российский технологический университет»**

ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

## **Лабораторная работа 2**

по курсу «Случайные процессы»

**ВАРИАНТ 6**

Тема: **Процесс роста и мутации популяции**

Выполнил:  
Студент 4-го курса  
Едренников Д.А.

Группа: КМБО-01-20

МОСКВА – 2023

## Задание

В популяции могут находиться объекты двух видов: N-объекты и M-объекты.

Дано:

- время жизни каждого N-объекта является случайной величиной, имеющей показательное распределение с параметром  $\lambda(t_b) = 0.005 * N(t_b-0) + 0.01 * M(t_b-0) + 0.3$  (где  $t_b$  – время рождения объекта,  $N(t_b-0)$  – число N-объектов до момента времени  $t_b$ ,  $M(t_b-0)$  – число M-объектов до момента времени  $t_b$ );

- время жизни каждого M-объекта является случайной величиной, имеющей показательное распределение с параметром  $\mu(t_b) = 0.05 * N(t_b-0) + 0.03 * M(t_b-0)$  (где  $t_b$  – время рождения объекта,  $N(t_b-0)$  – число N-объектов до момента времени  $t_b$ ,  $M(t_b-0)$  – число M-объектов до момента времени  $t_b$ );

- по окончании времени жизни каждый N-объект порождает с вероятностью  $pn_1$  один N-объект (событие  $S_N(1)$ ), с вероятностью  $pn_2$  два N-объекта (событие  $S_N(2)$ ), с вероятностью  $pn_{11} = 1 - pn_1 - pn_2$  один N-объект и один M-объект (событие  $S_N(3)$ );

- по окончании времени жизни каждый M-объект порождает с вероятностью  $pm_1$  один N-объект (событие  $S_M(1)$ ), ничего не порождает с вероятностью  $pm_0$  (событие  $S_M(0)$ ), с вероятностью  $pm_{11} = 1 - pm_1 - pm_0$  один N-объект и один M-объект (событие  $S_M(2)$ );

- до начального момента  $t = 0$  не было объектов, в начальный момент происходит событие  $S_N(1)$  и появляется первый объект: N-объект.

Состояние системы в момент времени  $t$  характеризуется параметрами  $(N(t), M(t))$ , где  $N(t)$  – число N-объектов,  $M(t)$  – число M-объектов. Событием в развитии системы называется момент окончания жизни (исчезновения) любого из объектов и (одновременно) появления новых объектов. События могут быть 6 типов:  $S_N(1)$ ,  $S_N(2)$ ,  $S_N(3)$ ,  $S_M(0)$ ,  $S_M(1)$ ,  $S_M(2)$ . При появлении каждого нового объекта случайным образом в соответствии с заданным законом распределения определяется время его жизни. Считать для первого события: момент наступления события  $t_{cob}(1) = 0$ ; тип события  $Type(1) = S_N(1)$ .

**Требуется:**

- 1) Провести моделирование первых 100 событий в развитии системы.
- 2) Составить следующие таблицы в соответствии с **Указаниями**.

Таблица 1 с данными о событиях:

- номер события  $i$ ;
- момент наступления события  $t_{cob}(i)$ ;
- тип события  $Type(i)$ ;
- время жизни появившихся новых объектов (2 столбца)  $t_{ж1}(i)$ ,  $t_{ж2}(i)$ ;
- состояние системы после события  $C(i)$ ;
- время ожидания до следующего события  $t_{ож}(i)$ ;

- номер  $J_{\text{кж}}(i)$  объекта, у которого раньше закончится жизнь;
- вид этого исчезающего объекта  $\text{Gen}_{\text{кж}}(i)$  (N или M).

Таблица 2 с данными об объектах:

- номер объекта  $j$ ;
- вид объекта  $\text{Gen}(j)$  (N или M)
- момент появления (рождения) объекта  $t_b(j)$ ;
- время жизни объекта  $t_l(j)$ ;
- момент исчезновения объекта  $t_d(j)$ ;
- номера объектов-потомков (2 столбца)  $\text{Des}_1(j)$ ,  $\text{Des}_2(j)$ .

Таблица 3 с данными о типах событий.

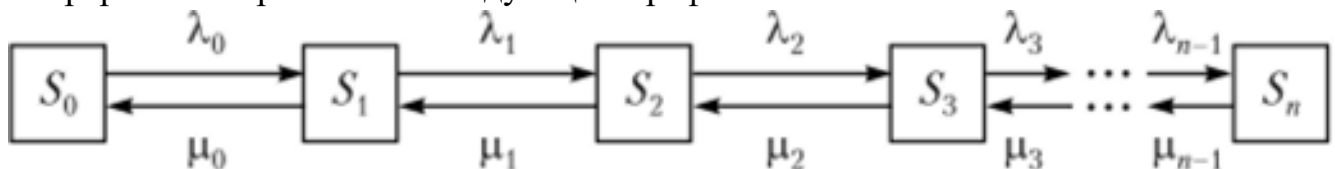
Таблица 4 с данными о видах объектов.

Таблица 5 с данными о состояниях, которые появились при моделировании и имеются в таблице 1.

Результаты в отчете приводить с округлением до 0.00001.

## Краткие теоретические сведения

Процессом рождения и гибели называется марковский процесс с непрерывным временем и следующим графом состояний:



Плотности  $\lambda_i$  ( $i = 0, 1, 2, \dots$ ) называются «интенсивностями размножения», а плотности  $\mu_j$  ( $j = 0, 1, 2, \dots$ ) – «интенсивностями гибели».

Дифференциальные уравнения Колмогорова:

$$\left\{ \begin{array}{l} \frac{dp_0(t)}{dt} = \lambda_0 p_0(t) + \mu_1 p_1(t) \\ \dots \\ \frac{dp_k(t)}{dt} = -(\lambda_k + \mu_k) p_k(t) + \lambda_{k-1} p_{k-1}(t) + \mu_{k+1} p_{k+1}(t), k = 1, 2, \dots, n-1 \\ \dots \\ \frac{dp_n(t)}{dt} = -\mu_n p_n(t) + \lambda_{n-1} p_{n-1}(t) \end{array} \right.$$

Формулы для нахождения стационарного распределения:

$$r_0 = (1 + \frac{\lambda_0}{\mu_1} + \dots + \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \dots \mu_n})^{-1}$$

$$r_k = \frac{\lambda_0 \lambda_1 \dots \lambda_{k-1}}{\mu_1 \dots \mu_k} r_0$$

Используемые функции из языка python:

expon.rvs(scale=1 /u) – генерации случайного значения из экспоненциального распределения.

random.random() – генерация случайного числа от 0 до 1.

## Результаты расчетов

Вариант равен 6.

pn <sub>1</sub>	pn <sub>2</sub>	Pn <sub>11</sub>	pm <sub>0</sub>	pm <sub>1</sub>	pm <sub>11</sub>
0.354	0.315	0.331	0.525	0.299	0.176

Таблица 1:

№ события	t <sub>собр</sub>	Type	t <sub>ж1</sub>	t <sub>ж2</sub>	C	t <sub>ож</sub>	J <sub>кж</sub>	Ge n <sub>кж</sub>
1	0	S <sub>N</sub> (1)	0.90605	-1.	( 1,0)	0.90605	1	N
2	0.90605	S <sub>N</sub> (1)	10.81899	-1.	( 1, 0)	10.81899	2	N
3	11.72504	S <sub>N</sub> (3)	0.48189	5.92856	( 1,1)	0.48189	3	N
4	12.20693	S <sub>N</sub> (2)	0.89708	8.78615	( 2,1)	0.89708	5	N
5	13.10401	S <sub>N</sub> (1)	2.54401	-1.	( 2, 1)	2.54401	7	N
6	15.64802	S <sub>N</sub> (2)	5.78234	9.43513	( 3, 1)	2.00557	4	M
7	17.6536	S <sub>M</sub> (0)	-1.	-1.	( 3, 0)	3.33949	6	N
8	20.99308	S <sub>N</sub> (3)	0.24487	1.95362	( 3, 1)	0.24487	10	N
9	21.23795	S <sub>N</sub> (1)	1.0672	-1.	( 3, 1)	0.19241	8	N
10	21.43036	S <sub>N</sub> (2)	0.06656	0.84906	( 4, 1)	0.06656	13	N
11	21.49692	S <sub>N</sub> (1)	0.13475	-1.	( 4, 1)	0.13475	15	N
12	21.63166	S <sub>N</sub> (3)	0.36333	7.32272	( 4, 2)	0.36333	16	M
13	21.99499	S <sub>M</sub> (1)	0.05621	-1.	( 4, 2)	0.05621	18	M
14	22.0512	S <sub>M</sub> (1)	1.71971	-1.	( 4, 2)	0.22822	14	N

15	22.27942	$S_N(1)$	2.61414	-1.	( 4, 2)	0.02573	12	N
16	22.30515	$S_N(2)$	0.11982	2.6037	( 5, 2)	0.11982	21	N
17	22.42498	$S_N(2)$	1.79462	1.98755	( 6, 2)	0.52173	11	M
18	22.94671	$S_M(2)$	9.7469	13.94962	( 7, 2)	0.82421	19	M
19	23.77091	$S_M(1)$	0.5608	-1.	( 7, 2)	0.44868	23	N
20	24.21959	$S_N(1)$	7.56766	-1.	( 7, 2)	0.11212	27	M
21	24.33171	$S_M(1)$	1.02508	-1.	( 7, 2)	0.08081	24	N
22	24.41253	$S_N(1)$	1.57986	-1.	( 7, 2)	0.48103	20	N
23	24.89356	$S_N(3)$	0.89837	0.9341	( 7, 3)	0.01529	22	N
24	24.90885	$S_N(2)$	4.80591	7.66686	( 8, 3)	0.1743	9	N
25	25.08315	$S_N(3)$	0.11422	1.52828	( 8, 4)	0.11422	35	N
26	25.19737	$S_N(1)$	4.56702	-1.	( 8, 4)	0.15943	29	M
27	25.3568	$S_M(1)$	6.79438	-1.	( 8, 4)	0.43513	31	N
28	25.79193	$S_N(2)$	0.17837	0.60284	( 9, 4)	0.03573	32	M
29	25.82766	$S_M(2)$	1.09605	2.98787	(10, 4)	0.14264	39	N
30	25.97029	$S_n(1)$	2.76908	-1.	(10, 4)	0.02209	30	N
31	25.99239	$S_n(1)$	8.19557	-1.	(10, 4)	0.40238	40	N
32	26.39477	$S_n(1)$	0.87901	-1.	(10, 4)	0.21666	36	M
33	26.61143	$S_M(2)$	0.80538	6.74821	(11, 4)	0.31228	41	N
34	26.92371	$S_N(3)$	1.16568	3.42029	(11, 5)	0.35007	45	N
35	27.27378	$S_N(2)$	2.39462	5.66188	(12, 5)	0.14302	46	M
36	27.4168	$S_M(1)$	0.13775	-1.	(12, 5)	0.13775	52	M
37	27.55455	$S_M(0)$	-1.	-1.	(12, 4)	0.53483	48	M
38	28.08939	$S_M(2)$	1.16467	4.57897	(13, 4)	0.64999	43	N
39	28.73938	$S_N(2)$	0.15692	1.4944	(14, 4)	0.07615	42	M
40	28.81553	$S_M(1)$	0.2466	-1.	(14, 4)	0.08077	55	N
41	28.8963	$S_N(2)$	1.31255	2.06901	(15, 4)	0.05808	17	N
42	28.95438	$S_N(2)$	0.07469	0.70806	(16, 4)	0.07469	60	N
43	29.02907	$S_N(2)$	1.46181	6.17155	(17, 4)	0.03305	57	M
44	29.06213	$S_M(0)$	-1.	-1.	(17, 3)	0.19193	53	M
45	29.25405	$S_M(1)$	0.91265	-1.	(17, 3)	0.40839	61	N
46	29.66244	$S_N(3)$	0.47541	2.65026	(17, 4)	0.00596	50	N
47	29.6684	$S_N(3)$	1.11961	2.38917	(17, 5)	0.04636	33	N
48	29.71476	$S_n(1)$	7.7008	-1.	(17, 5)	0.04963	37	N
49	29.76439	$S_N(3)$	0.7565	1.08709	(17, 6)	0.37346	65	M
50	30.13785	$S_M(0)$	-1.	-1.	(17, 5)	0.02885	64	M
51	30.1667	$S_M(1)$	1.23748	-1.	(17, 5)	0.04214	58	N

52	30.20885	$S_N(2)$	0.67139	2.5596	(18, 5)	0.02493	56	N
53	30.23377	$S_N(2)$	0.41257	0.53026	(19, 5)	0.11022	49	N
54	30.344	$S_N(2)$	0.13162	0.90537	(20, 5)	0.13162	77	N
55	30.47562	$S_n(1)$	0.45445	-1.	(20, 5)	0.01527	62	N
56	30.49089	$S_N(2)$	1.63826	3.0284	(21, 5)	0.03	70	M
57	30.52089	$S_M(0)$	-1.	-1.	(21, 4)	0.12546	75	N
58	30.64634	$S_N(3)$	2.43577	2.73866	(21, 5)	0.11769	76	N
59	30.76404	$S_N(2)$	0.26508	1.74643	(22, 5)	0.02397	67	M
60	30.78801	$S_M(1)$	0.24703	-1.	(22, 5)	0.06347	71	N
61	30.85148	$S_N(3)$	0.79786	0.98267	(22, 6)	0.02877	73	N
62	30.88024	$S_N(2)$	1.91289	3.39659	(23, 6)	0.04983	79	N
63	30.93007	$S_N(3)$	0.08033	1.1329	(23, 7)	0.03524	59	N
64	30.9653	$S_n(1)$	1.58152	-1.	(23, 7)	0.04509	91	M
65	31.01039	$S_M(2)$	0.10823	3.97145	(24, 7)	0.01873	84	N
66	31.02912	$S_n(1)$	3.65856	-1.	(24, 7)	0.00592	86	M
67	31.03504	$S_M(2)$	0.44836	1.77214	(25, 7)	0.08359	94	M
68	31.11863	$S_M(1)$	0.43389	-1.	(25, 7)	0.13074	78	N
69	31.24937	$S_N(2)$	0.76331	1.72003	(26, 7)	0.15481	72	M
70	31.40418	$S_M(0)$	-1.	-1.	(26, 6)	0.07922	97	M
71	31.4834	$S_M(1)$	0.16342	-1.	(26, 6)	0.06912	99	M
72	31.55252	$S_M(1)$	0.6004	-1.	(26, 6)	0.0943	102	M
73	31.64682	$S_M(1)$	0.47539	-1.	(26, 6)	0.00252	87	N
74	31.64934	$S_n(1)$	1.45347	-1.	(26, 6)	0.13792	28	N
75	31.78726	$S_N(2)$	0.17673	4.49805	(27, 6)	0.04689	88	M
76	31.83414	$S_M(0)$	-1.	-1.	(27, 5)	0.12985	106	N
77	31.96399	$S_n(1)$	0.09037	-1.	(27, 5)	0.0487	100	N
78	32.01268	$S_n(1)$	1.75533	-1.	(27, 5)	0.04167	108	N
79	32.05436	$S_N(3)$	1.20688	1.51647	(27, 6)	0.00321	68	N
80	32.05757	$S_n(1)$	0.05345	-1.	(27, 6)	0.00539	92	N
81	32.06296	$S_n(1)$	4.44639	-1.	(27, 6)	0.04805	112	N
82	32.11102	$S_N(2)$	0.2811	2.5665	(28, 6)	0.01119	104	M
83	32.12221	$S_M(0)$	-1.	-1.	(28, 5)	0.00694	80	N
84	32.12914	$S_N(3)$	0.57365	1.87121	(28, 6)	0.02204	38	M
85	32.15118	$S_M(0)$	-1.	-1.	(28, 5)	0.00174	103	M
86	32.15292	$S_M(1)$	1.44544	-1.	(28, 5)	0.15977	66	N
87	32.3127	$S_n(1)$	1.43666	-1.	(28, 5)	0.07942	114	N
88	32.39212	$S_N(3)$	0.14112	1.38567	(28, 6)	0.11835	85	N

89	32.51046	$S_n(1)$	7.36214	-1.	(28, 6)	0.02277	120	M
90	32.53324	$S_M(0)$	-1.	-1.	(28, 5)	0.01358	93	N
91	32.54682	$S_n(1)$	7.31561	-1.	(28, 5)	0.02889	34	N
92	32.57571	$S_n(1)$	1.18008	-1.	(28, 5)	0.09264	54	N
93	32.66835	$S_n(1)$	0.56752	-1.	(28, 5)	0.02526	25	N
94	32.69361	$S_n(1)$	1.22673	-1.	(28, 5)	0.00919	116	M
95	32.7028	$S_M(1)$	0.16457	-1.	(28, 5)	0.06564	74	N
96	32.76844	$S_N(2)$	0.64568	0.74179	(29, 5)	0.02468	89	N
97	32.79313	$S_N(3)$	0.18215	0.9681	(29, 6)	0.01406	98	N
98	32.80719	$S_N(3)$	0.70849	2.33071	(29, 7)	0.06018	127	M
99	32.86737	$S_M(2)$	1.25868	2.2532	(30, 7)	0.0683	51	N
100	32.93566	$S_n(1)$	0.4769	-1.	(30, 7)	0.03374	101	N

Таблица 2:

№ объекта	Gen	$t_b$	$t_l$	$t_d$	Des <sub>1</sub>	Des <sub>2</sub>
1	N	0	0.90605	0.90605	2	-1
2	N	0.90605	10.81899	11.72504	3	4
3	N	11.72504	0.48189	12.20693	5	6
4	M	11.72504	5.92856	17.6536	-1	-1
5	N	12.20693	0.89708	13.10401	7	-1
6	M	12.20693	8.78615	20.99308	10	11
6	N	13.10401	2.54401	15.64802	8	9
8	N	15.64802	5.78234	21.43036	13	14
9	M	15.64802	9.43513	25.08315	35	36
10	N	20.99308	0.24487	21.23795	12	-1
11	M	20.99308	1.95362	22.94671	25	26
11	N	21.23795	1.0672	22.30515	21	22
13	N	21.43036	0.06656	21.49692	15	-1
14	M	21.43036	0.84906	22.27942	20	-1
14	N	21.49692	0.13475	21.63166	16	17
16	M	21.63166	0.36333	21.99499	18	-1
17	N	21.63166	7.32272	28.95438	60	61

18	M	21.99499	0.05621	22.0512	19	-1
19	M	22.0512	1.71971	23.77091	27	-1
19	N	22.27942	2.61414	24.89356	31	32
21	N	22.30515	0.11982	22.42498	23	24
22	M	22.30515	2.6037	24.90885	33	34
23	N	22.42498	1.79462	24.21959	28	-1
24	M	22.42498	1.98755	24.41253	30	-1
25	N	22.94671	9.7469	32.69361	126	-1
26	M	22.94671	13.94962	36.89633	-	-
27	M	23.77091	0.5608	24.33171	29	-1
27	N	24.21959	7.56766	31.78726	106	107
29	M	24.33171	1.02508	25.3568	38	-1
29	N	24.41253	1.57986	25.99239	44	-1
31	N	24.89356	0.89837	25.79193	39	40
32	M	24.89356	0.9341	25.82766	41	42
33	N	24.90885	4.80591	29.71476	69	-1
34	M	24.90885	7.66686	32.57571	124	-1
35	N	25.08315	0.11422	25.19737	37	-1
36	M	25.08315	1.52828	26.61143	46	47
36	N	25.19737	4.56702	29.76439	70	71
38	M	25.3568	6.79438	32.15118	-1	-1
39	N	25.79193	0.17837	25.97029	43	-1
40	M	25.79193	0.60284	26.39477	45	-1
41	N	25.82766	1.09605	26.92371	48	49
42	M	25.82766	2.98787	28.81553	57	-1
42	N	25.97029	2.76908	28.73938	55	56
43	N	25.99239	8.19557	34.18796	-	-
44	N	26.39477	0.87901	27.27378	50	51
46	M	26.61143	0.80538	27.4168	52	-1
47	N	26.61143	6.74821	33.35964	-	-
48	M	26.92371	1.16568	28.08939	53	54
49	N	26.92371	3.42029	30.344	77	78
50	N	27.27378	2.39462	29.6684	67	68
51	M	27.27378	5.66188	32.93566	136	-1
52	M	27.4168	0.13775	27.55455	-1	-1
53	M	28.08939	1.16467	29.25405	64	-1



54	N	28.08939	4.57897	32.66835	125	-1
55	N	28.73938	0.15692	28.8963	58	59
56	M	28.73938	1.4944	30.23377	75	76
57	M	28.81553	0.2466	29.06213	-1	-1
58	N	28.8963	1.31255	30.20885	73	74
59	M	28.8963	2.06901	30.9653	93	-1
60	N	28.95438	0.07469	29.02907	62	63
61	M	28.95438	0.70806	29.66244	65	66
62	N	29.02907	1.46181	30.49089	80	81
63	M	29.02907	6.17155	35.20063	-	-
64	M	29.25405	0.91265	30.1667	72	-1
65	M	29.66244	0.47541	30.13785	-1	-1
66	N	29.66244	2.65026	32.3127	119	-1
67	M	29.6684	1.11961	30.78801	86	-1
68	N	29.6684	2.38917	32.05757	112	-1
68	N	29.71476	7.7008	37.41556	-	-
70	M	29.76439	0.7565	30.52089	-1	-1
71	N	29.76439	1.08709	30.85148	87	88
72	M	30.1667	1.23748	31.40418	-1	-1
73	N	30.20885	0.67139	30.88024	89	90
74	M	30.20885	2.5596	32.76844	128	129
75	N	30.23377	0.41257	30.64634	82	83
76	M	30.23377	0.53026	30.76404	84	85
77	N	30.344	0.13162	30.47562	79	-1
78	M	30.344	0.90537	31.24937	100	101
78	N	30.47562	0.45445	30.93007	91	92
80	N	30.49089	1.63826	32.12914	116	117
81	M	30.49089	3.0284	33.51929	-	-
82	M	30.64634	2.43577	33.08211	-	-
83	N	30.64634	2.73866	33.385	-	-
84	N	30.76404	0.26508	31.02912	96	-1
85	M	30.76404	1.74643	32.51046	122	-1
86	M	30.78801	0.24703	31.03504	97	98
87	N	30.85148	0.79786	31.64934	105	-1
88	M	30.85148	0.98267	31.83414	-1	-1
89	N	30.88024	1.91289	32.79313	130	131

90	M	30.88024	3.39659	34.27684	-	-
91	M	30.93007	0.08033	31.01039	94	95
92	N	30.93007	1.1329	32.06296	113	-1
92	N	30.9653	1.58152	32.54682	123	-1
94	M	31.01039	0.10823	31.11863	99	-1
95	N	31.01039	3.97145	34.98184	-	-
95	N	31.02912	3.65856	34.68768	-	-
97	M	31.03504	0.44836	31.4834	102	-1
98	N	31.03504	1.77214	32.80719	132	133
99	M	31.11863	0.43389	31.55252	103	-1
100	N	31.24937	0.76331	32.01268	109	-1
101	M	31.24937	1.72003	32.9694	-	-
102	M	31.4834	0.16342	31.64682	104	-1
103	M	31.55252	0.6004	32.15292	118	-1
104	M	31.64682	0.47539	32.12221	-1	-1
104	N	31.64934	1.45347	33.10281	-	-
106	N	31.78726	0.17673	31.96399	108	-1
107	M	31.78726	4.49805	36.28531	-	-
107	N	31.96399	0.09037	32.05436	110	111
108	N	32.01268	1.75533	33.76802	-	-
110	N	32.05436	1.20688	33.26123	-	-
111	M	32.05436	1.51647	33.57082	-	-
111	N	32.05757	0.05345	32.11102	114	115
112	N	32.06296	4.44639	36.50936	-	-
114	N	32.11102	0.2811	32.39212	120	121
115	M	32.11102	2.5665	34.67752	-	-
116	M	32.12914	0.57365	32.7028	127	-1
117	N	32.12914	1.87121	34.00036	-	-
118	M	32.15292	1.44544	33.59836	-	-
118	N	32.3127	1.43666	33.74936	-	-
120	M	32.39212	0.14112	32.53324	-1	-1
121	N	32.39212	1.38567	33.77778	-	-
121	N	32.51046	7.36214	39.8726	-	-
122	N	32.54682	7.31561	39.86243	-	-
123	N	32.57571	1.18008	33.75579	-	-
124	N	32.66835	0.56752	33.23587	-	-

125	N	32.69361	1.22673	33.92034	-	-
127	M	32.7028	0.16457	32.86737	134	135
128	N	32.76844	0.64568	33.41412	-	-
129	M	32.76844	0.74179	33.51024	2	-1
130	M	32.79313	0.18215	32.97528	3	4
131	N	32.79313	0.9681	33.76123	5	6
132	M	32.80719	0.70849	33.51568	-1	-1
133	N	32.80719	2.33071	35.1379	7	-1
134	M	32.86737	1.25868	34.12605	10	11
135	N	32.86737	2.2532	35.12056	8	9
135	N	32.93566	0.4769	33.41256	13	14

Таблица 3:

Тип события	$S_N(1)$	$S_N(2)$	$S_N(3)$	$S_M(0)$	$S_M(1)$	$S_M(2)$	
Число событий	28	22	17	10	16	7	100
Относительная частота	0.28	0.22	0.17	0.1	0.16	0.07	1.0

Таблица 4:

Вид объекта	Число появившихся объектов за время $[0, t_{\text{cob}}(100)]$	Число объектов в момент $t_{\text{cob}}(100)$
N	96	30
M	40	7

Таблица 5:

Состояние	$n_{\text{сост}}$	$v_{\text{сост}}$	$T_{\text{сост}}$	$\Delta_{\text{сост}}$
(1, 0)	2	0.02	11.72504	0.356
(1, 1)	1	0.01	0.48189	0.01463
(2, 1)	2	0.02	3.44109	0.10448
(3, 0)	1	0.01	3.33949	0.10139

(3, 1)	3	0.03	2.44285	0.07417
(4, 1)	2	0.02	0.2013	0.00611
(4, 2)	4	0.04	0.67349	0.02045
(5, 2)	1	0.01	0.11982	0.00364
(6, 2)	1	0.01	0.52173	0.01584
(7, 2)	5	0.05	1.94685	0.05911
(7, 3 )	1	0.01	0.01529	0.00046
(8, 3)	1	0.01	0.1743	0.00529
(8, 4)	3	0.03	0.70878	0.02152
(9, 4 )	1	0.01	0.03573	0.00108
(10, 4)	4	0.04	0.78377	0.0238
(11, 4)	1	0.01	0.31228	0.00948
(11, 5)	1	0.01	0.35007	0.01063
(12, 4)	1	0.01	0.53483	0.01624
(12, 5)	2	0.02	0.28077	0.00852
(13, 4)	1	0.01	0.64999	0.01974
(14, 4)	2	0.02	0.15692	0.00476
(15, 4)	1	0.01	0.05808	0.00176
(16, 4)	1	0.01	0.07469	0.00227
(17, 3)	2	0.02	0.60031	0.01823
(17, 4)	2	0.02	0.03901	0.00118
(17, 5)	4	0.04	0.16698	0.00507
(17, 6)	1	0.01	0.37346	0.01134
(18, 5)	1	0.01	0.02493	0.00076
(19, 5)	1	0.01	0.11022	0.00335
(20, 5 )	2	0.02	0.14689	0.00446
(21, 4)	1	0.01	0.12546	0.00381
(21, 5)	2	0.02	0.14769	0.00448
(22, 5)	2	0.02	0.08744	0.00265
(22, 6)	1	0.01	0.02877	0.00087
(23, 6)	1	0.01	0.04983	0.00151
(23, 7)	2	0.02	0.08033	0.00244
(24, 7)	2	0.02	0.02465	0.00075
(25, 7)	2	0.02	0.21433	0.00651
(26, 6)	5	0.05	0.38307	0.01163
(26,7)	1	0.01	0.15481	0.0047

(27, 5)	3	0.03	0.22022	0.00669
(27, 6)	4	0.04	0.10354	0.00314
(28, 5)	10	0.1	0.48308	0.01467
(28, 6)	4	0.04	0.17435	0.00529
(29, 5)	1	0.01	0.02468	0.00075
(29, 6)	1	0.01	0.01406	0.00043
(29, 7)	1	0.01	0.06018	0.00183
(30, 7)	2	0.02	0.0683	0.00207
	100	1.0	32.93566	1.0

### **Список литературы**

1. Случайные процессы [Электронный ресурс]: методические указания / А. А. Лобузов. — М.: РТУ МИРЭА, 2021 — 36с.
2. Вентцель Е.С., Овчаров Л.А. Теория случайных процессов и её инженерные приложения. - М.: Кнорус, 2018 - 488 с.
3. Бородин А.Н. Случайные процессы. - СПб.: Лань, 2021 - 640с.

## Приложение

```
import numpy as np
import random
from scipy.stats import expon

f = open('answer.txt', 'r+')
np.set_printoptions(suppress=True)

pn1 = 0.354
pn2 = 0.315
pn3 = 1 - pn1 - pn2
pm0 = 0.525
pm1 = 0.299
pm2 = 1 - pm0 - pm1

N = 0
M = 0

L = list(range(1, 101))

Ttime = []
Ttype = []
Tl1 = []
Tl2 = []
condition = []
Tremained = []
Tdin = []
TdG = []

jN = np.zeros(300)
jG = np.zeros(300)
jB = np.zeros(300)
jLt = np.zeros(300)
jD = np.zeros(300)
jS1 = np.zeros(300)
jS2 = np.zeros(300)

Sn1 = 0
Sn2 = 0
Sn3 = 0

Sm0 = 0
Sm1 = 0
Sm2 = 0
```

```
Sn1d = 0
Sn2d = 0
Sn3d = 0
Sm0d = 0
Sm1d = 0
Sm2d = 0
```

```
n5 = [[0 for j in range(15)] for j in range(50)]
v5 = [[0 for j in range(15)] for j in range(50)]
T5 = [[0 for j in range(15)] for j in range(50)]
D5 = [[0 for j in range(15)] for j in range(50)]
```

```
Ld = []
Ldi = []
```

```
i = 1
```

```
Ttime.append(0)
Ttype.append("Sn1")
Tl1.append(expon.rvs(scale=1 / 0.3))
N += 1
Tl2.append(-1)
condition.append((1, 0))
Ld.append(Tl1[0])
Ldi.append("N")
Tremained.append(Tl1[0])
Tdin.append(i)
TdG.append("N")
```

```
jN[0] = i
jG[0] = 1
jB[0] = 0
jLt[0] = Tl1[0]
jD[0] = Tl1[0]
```

```
Sn1 += 1
n5[1][0] += 1
T5[1][0] += Tl1[0]
# T1 += service_time[i]
```



```

while len(Ttime) != 100:
    if Ldi[np.argmin(Ld)] == "N":

        Ttime.append(Ttime[-1] + Ld[np.argmin(Ld)])
        c = random.random()
        if c < pn1:
            jS1[np.argmin(Ld)] = i + 1
            jS2[np.argmin(Ld)] = -1

            Ttype.append("Sn1")
            Tl1.append(expon.rvs(scale=1 / (0.005 * N + 0.01 * M + 0.3)))
            Tl2.append(-1)
            condition.append(condition[-1])
            Ld -= Ld[np.argmin(Ld)]
            Ld[np.argmin(Ld)] = (float("inf"))
            Ld = np.append(Ld, Tl1[-1])
            Ldi.append("N")
            Tremained.append(Ld[np.argmin(Ld)])
            Tdin.append(np.argmin(Ld)+1)
            TdG.append(Ldi[np.argmin(Ld)])

        jN[i] = i
        jG[i] = 1
        jB[i] = Ttime[-1]
        jLt[i] = Ld[i]
        jD[i] = Ttime[-1] + Ld[i]
        i += 1
        Sn1 += 1
        n5[N][M] = n5[N][M] + 1
        T5[N][M] += Tremained[-1]
    elif c < pn2 + pn1:
        jS1[np.argmin(Ld)] = i + 1
        jS2[np.argmin(Ld)] = i + 2

        Ttype.append("Sn2")
        a = expon.rvs(scale=1 / (0.005 * N + 0.01 * M + 0.3))
        b = expon.rvs(scale=1 / (0.005 * N + 0.01 * M + 0.3))
        if b > a:
            Tl1.append(a)
            Tl2.append(b)
        else:
            Tl1.append(b)

```

```

        Tl2.append(a)
        condition.append((condition[-1][0] + 1, condition[-1][1]))
        Ld -= Ld[np.argmin(Ld)]
        Ld[np.argmin(Ld)] = (float("inf"))
        Ld = np.append(Ld, Tl1[-1])
        Ldi.append("N")
        Ld = np.append(Ld, Tl2[-1])
        Ldi.append("N")
        Tremained.append(Ld[np.argmin(Ld)])
        Tdin.append(np.argmin(Ld)+1)
        TdG.append(Ldi[np.argmin(Ld)])

jN[i] = i + 1
jG[i] = 1
jB[i] = Ttime[-1]
jLt[i] = Ld[i]
jD[i] = Ttime[-1] + Ld[i]
i += 1

jN[i] = i + 1
jG[i] = 2
jB[i] = Ttime[-1]
jLt[i] = Ld[i]
jD[i] = Ttime[-1] + Ld[i]
i += 1

Sn2 += 1
N += 1
n5[N][M] += 1
T5[N][M] += Tremained[-1]
else:
    Ttype.append("Sn3")
    Sn3+=1

jS1[np.argmin(Ld)] = i + 1
jS2[np.argmin(Ld)] = i + 2

a = expon.rvs(scale=1 / (0.005 * N + 0.01 * M + 0.3))
b = expon.rvs(scale=1 / (0.05 * N + 0.03 * M))
Ld -= Ld[np.argmin(Ld)]
Ld[np.argmin(Ld)] = (float("inf"))

```

```

if b > a:
    Tl1.append(a)
    Tl2.append(b)
    Ld = np.append(Ld, Tl1[-1])
    Ldi.append("N")
    Ld = np.append(Ld, Tl2[-1])
    Ldi.append("M")

    jN[i] = i + 1
    jG[i] = 1
    jB[i] = Ttime[-1]
    jLt[i] = Ld[i]
    jD[i] = Ttime[-1] + Ld[i]
    i += 1

    jN[i] = i + 1
    jG[i] = 2
    jB[i] = Ttime[-1]
    jLt[i] = Ld[i]
    jD[i] = Ttime[-1] + Ld[i]
    i += 1
else:
    Tl1.append(b)
    Tl2.append(a)
    Ld = np.append(Ld, (Tl1[-1]))
    Ldi.append("M")
    Ld = np.append(Ld, (Tl2[-1]))
    Ldi.append("N")

    jN[i] = i + 1
    jG[i] = 2
    jB[i] = Ttime[-1]
    jLt[i] = Ld[i]
    jD[i] = Ttime[-1] + Ld[i]
    i += 1

    jN[i] = i + 1
    jG[i] = 1
    jB[i] = Ttime[-1]
    jLt[i] = Ld[i]
    jD[i] = Ttime[-1] + Ld[i]
    i += 1

```

```

condition.append((condition[-1][0], condition[-1][1] + 1))
Tremained.append(Ld[np.argmin(Ld)])
Tdin.append(np.argmin(Ld)+1)
TdG.append(Ldi[np.argmin(Ld)])
M += 1
n5[N][M] += 1
T5[N][M] += Tremained[-1]
else:
    Time.append(Ttime[-1] + Ld[np.argmin(Ld)])
    c = random.random()
    if c < pn1:
        Ttype.append("Sm1")

        jS1[np.argmin(Ld)] = i + 1
        jS2[np.argmin(Ld)] = -1

        Tl1.append(expon.rvs(scale=1 / (0.05 * N + 0.03 * M)))
        Tl2.append(-1)
        condition.append(condition[-1])
        Ld -= Ld[np.argmin(Ld)]
        Ld[np.argmin(Ld)] = (float("inf"))
        Ld = np.append(Ld, (Tl1[-1]))
        Ldi.append("M")
        Tremained.append(Ld[np.argmin(Ld)])
        Tdin.append(np.argmin(Ld) + 1)
        TdG.append(Ldi[np.argmin(Ld)])

        jN[i] = i + 1
        jG[i] = 2
        jB[i] = Ttime[-1]
        jLt[i] = Ld[i]
        jD[i] = Ttime[-1] + Ld[i]
        i += 1

        Sm1 += 1
        n5[N][M] += 1
        T5[N][M] += Tremained[-1]
    elif c < pn2 + pn1:
        Ttype.append("Sm0")

        jS1[np.argmin(Ld)] = -1
        jS2[np.argmin(Ld)] = -1

```

```

Tl1.append(-1)
Tl2.append(-1)
condition.append((condition[-1][0], condition[-1][1] - 1))
Ld -= Ld[np.argmin(Ld)]
Ld[np.argmin(Ld)] = (float("inf"))
Tremained.append(Ld[np.argmin(Ld)])
Tdin.append(np.argmin(Ld)+1)
TdG.append(Ldi[np.argmin(Ld)])

Sm0 += 1
M -= 1
n5[N][M] += 1
T5[N][M] += Tremained[-1]
else:
    Ttype.append("Sm2")

jS1[np.argmin(Ld)] = i + 1
jS2[np.argmin(Ld)] = i + 2

a = expon.rvs(scale=1 / (0.005 * N + 0.01 * M + 0.3))
b = expon.rvs(scale=1 / (0.05 * N + 0.03 * M))
Ld -= Ld[np.argmin(Ld)]
Ld[np.argmin(Ld)] = (float("inf"))

if b > a:
    Tl1.append(a)
    Tl2.append(b)
    Ld = np.append(Ld, Tl1[-1])
    Ldi.append("N")
    Ld = np.append(Ld, (Tl2[-1]))
    Ldi.append("M")

jN[i] = i + 1
jG[i] = 1
jB[i] = Ttime[-1]
jLt[i] = Ld[i]
jD[i] = Ttime[-1] + Ld[i]
i += 1

jN[i] = i + 1
jG[i] = 2

```

```

    jB[i] = Ttime[-1]
    jLt[i] = Ld[i]
    jD[i] = Ttime[-1] + Ld[i]
    i += 1
else:
    Tl1.append(b)
    Tl2.append(a)
    Ld = np.append(Ld, (Tl1[-1]))
    Ldi.append("M")
    Ld = np.append(Ld, (Tl2[-1]))
    Ldi.append("N")

    jN[i] = i + 1
    jG[i] = 2
    jB[i] = Ttime[-1]
    jLt[i] = Ld[i]
    jD[i] = Ttime[-1] + Ld[i]
    i += 1

    jN[i] = i + 1
    jG[i] = 1
    jB[i] = Ttime[-1]
    jLt[i] = Ld[i]
    jD[i] = Ttime[-1] + Ld[i]
    i += 1

condition.append((condition[-1][0]+1, condition[-1][1]))
Tremained.append(Ld[np.argmin(Ld)])
Tdin.append(np.argmin(Ld)+1)
TdG.append(Ldi[np.argmin(Ld)])
Sm2 += 1
N += 1
n5[N][M] += 1
T5[N][M] += Tremained[-1]

Sn1d = Sn1 / (Sn1 + Sn2 + Sn3 + Sm1 + Sm0 + Sm2)
Sn2d = Sn2 / (Sn1 + Sn2 + Sn3 + Sm1 + Sm0 + Sm2)
Sn3d = Sn3 / (Sn1 + Sn2 + Sn3 + Sm1 + Sm0 + Sm2)
Sm0d = Sm0 / (Sn1 + Sn2 + Sn3 + Sm1 + Sm0 + Sm2)
Sm1d = Sm1 / (Sn1 + Sn2 + Sn3 + Sm1 + Sm0 + Sm2)
Sm2d = Sm2 / (Sn1 + Sn2 + Sn3 + Sm1 + Sm0 + Sm2)

T5[N][M] -= Tremained[-1]
v5 = np.array(n5) / 100

```

```
D5 = np.array(T5) / Ttime[-1]
```

```
Gn = 0
```

```
Gm = 0
```

```
for i in range(len(Ld)):
```

```
    if (Ldi[i] == "N"):
```

```
        Gn += 1
```

```
    else:
```

```
        Gm += 1
```

```
f.write(str(np.around(L, 5)))
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write(str(np.around(Ttime, 5)))
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write(str(Ttype))
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write(str(np.around(Tl1, 5)))
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write(str(np.around(Tl2, 5)))
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write(str(np.around(condition, 5)))
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write(str(np.around(Tremained, 5)))
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write(str(np.around(Tdin, 5)))
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write(str(TdG))
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write(str(np.around(jN, 5)))
```

```
f.write("\n")
```

```
f.write("\n")
```

```
f.write(str(np.around(jG, 5)))
```

```

f.write('\n')
f.write('\n')
f.write(str(np.around(jB, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jLt, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jD, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jS1, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(jS2, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(Sn1, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Sn2, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Sn3, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Sm0, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Sm1, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Sm2, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Sn1d, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Sn2d, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Sn3d, 5)))

```



```

f.write('\n')
f.write('\n')
f.write(str(np.around(Sm0d, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Sm1d, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Sm2d, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around((Sn1 + Sn2 + Sn3 + Sm1 + Sm0 + Sm2), 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around((Sn1d + Sn2d + Sn3d + Sm1d + Sm0d + Sm2d), 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(Gn, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(Gm, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(N, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(M, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write(str(np.around(n5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(v5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(T5, 5)))
f.write('\n')
f.write('\n')
f.write(str(np.around(D5, 5)))
f.write('\n')

```

```

f.write('\n')
total = 0
for ele in range(0, 50):
    total += sum(n5[ele])
f.write(str(np.around(total, 5)))
f.write('\n')
f.write('\n')
total = 0
for ele in range(0, 50):
    total += sum(v5[ele])
f.write(str(np.around(total, 5)))
f.write('\n')
f.write('\n')
total = 0
for ele in range(0, 50):
    total += sum(T5[ele])
f.write(str(np.around(total, 5)))
f.write('\n')
f.write('\n')
total = 0
for ele in range(0, 50):
    total += sum(D5[ele])
f.write(str(np.around(total, 5)))
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')
f.write('\n')

```