

بسم تعالی



شبیه سازی کامپیوتری

تمرین اول

استاد:

دکتر بردیا صفایی

نویسنده :

محمد هومان کشوری

شماره دانشجویی :

99105667

سوال 1.

(الف)

1. نیازمند یادگیری و بدست آوردن مهارت : گذاشتن زمان نسبتا کافی برای یادگیری
2. امکان دارد تحلیل نتایج شبیه سازی سخت باشد : شبیه سازی بر اساس یک الگوی مشخص تا حدی کار را ساده می کند
3. شبیه سازی می تواند زمانگیر و هزینه بر باشد که گاهی اوقات می توانیم آنرا کاهش دهیم.

(ب)

1. روش های تحلیلی نیازمند قلم و کاغذ هستند در صورتی که شبیه سازی نیازمند کامپیوتر است.
2. در سیستم های غیرخطی، شبیه سازی به ازای هر ورودی و فرض، می تواند نتایج جدیدی تولید کند.
3. در روش تحلیلی، جواب بدست آمده برروی تمام ورودی ها صادق است..
4. روش تحلیلی جواب های دقیق می دهد در صورتی که شبیه سازی جواب های تقریبی به ما می دهد.

(پ)

1. در کاربردهای نظامی، حتما نیاز به شبیه سازی است چرا که پارامترها باید ابتدا مورد آزمایش قرار بگیرند.
2. در مدل کردن های اقتصادی چرا که آینده را پیش بینی و شبیه سازی می کند.

سوال 2.

(الف)

موجودیت‌ها : کارمندان – مشتریان
ویژگی‌ها : تعداد درخواست‌ها
فعالیت‌ها : ارائه خدمات توسط یک کارمند به یک مشتری
رویدادها : ورود و خروج مشتری
متغیرهای حالت : تعداد کارمندان – تعداد مشتری‌ها

(ب)

موجودیت‌ها : مشتری‌ها - راننده‌ها
ویژگی‌ها : وضعیت مسافر - هزینه هر سفر
فعالیت‌ها : درخواست ماشین توسط مشتری
رویدادها : سوار شدن در تاکسی توسط مسافر و پیاده شدن از آن
متغیرهای حالت : تعداد مسافران و راننده‌ها

(پ)

موجودیت‌ها : کارمندان – مشتریان – کالاها
ویژگی‌ها : خرید مشتریان – زمان ورود مشتریان
فعالیت‌ها : زمان گشتن به دنبال کالای مورد نظر توسط مشتری
رویدادها : ورود و خروج مشتری جدید
متغیرهای حالت : تعداد کالاهای مورد نیاز مشتری – تعداد کارمندان فعال در صندوق

(ت)

موجودیت‌ها : کاربران - سرور
ویژگی‌ها : تعداد پیام‌ها – نقطه سکونت کاربران

فعالیت‌ها : زمان رسیدن درخواست ارسال پیام یک کاربر

رویدادها : ثبت نام کاربر – از کار افتادن سرور

متغیرهای حالت : تعداد کاربران در آن واحد

سوال 3.

از موارد کاربرد شبیه‌سازی در علوم کامپیوتر می‌توان به موارد زیر اشاره کرد :
شبیه‌سازی ترافیک شبکه – شبیه‌سازی سیستم‌عامل – شبیه‌سازی مدل‌های امنیتی

(الف)

فرض کنید می‌خواهیم یک سیستم کنترل لوازم خانگی هوشمند از راه دور طراحی کنیم. برای شبیه‌سازی این سیستم، باید تعدادی سنسور و دستگاه همراه نوع ارتباط آنها با یکدیگر و ویژگی بخصوص‌شان در سیستم قرار دهیم. سپس باید در شرایط مختلف سیستم را ارزیابی کنیم، مثلاً هنگامی که شرایط عادی است یا در مواقعی که برق می‌رود و یا در فصول مختلف سال. در نهایت با استفاده از نتایج این شبیه‌سازی می‌توانیم سیستم کنترل لوازم خانگی را از نظر مصرف انرژی بهینه کنیم و نیز بهترین تعداد سنسور و هزینه را برای این سیستم برآورد کنیم.

(ب)

در پردازش ابری فرض کنید می‌خواهیم یک سیستم اتوماسیون خدمات مبتنی بر زیرساخت ابری ارائه دهیم. در این سیستم باید سرورها و ویژگی‌های آنها مثل مکان جغرافیایی آنها، توان پردازشی، پهنای باند، سیستم‌عامل آنها و نیز موارد این چنین در نظر گرفته شود. سپس باید با سناریوهای مختلف این سیستم مورد ارزیابی قرار بگیرد، مثلاً اگر تعداد کاربران سیستم از تعدادی بالاتر رفت یا اینکه آگه یکی از سرورها دچار مشکل شد و یا اینکه اگر به یکی از سرورها حمله امنیتی شود.

در نهایت با استفاده از نتایج شبیه‌سازی می‌توان یک سیستم اتوماسیون طراحی کرد که از نظر توزیع بار بهینه عمل کند و نیز قابلیت پشتیبانی از تعداد قابل توجهی از کاربران را داشته باشد.

(ج)

در سیستم‌های بی درنگ، برای زمان‌بندی می‌توان از شبیه‌سازی استفاده کرد تا عملکرد سیستم را در شرایط بحرانی با مدت زمان پاسخ کوتاه بررسی کند. مثلاً فرض کنید یک سیستم هوشمند کنترل حادثه باری ماشین داریم که باید به صورت بی درنگ عمل کند به این منظور که اگر اشتباهی از سمت راننده داشته باشیم بتواند در کمترین زمان ممکن بهترین پاسخ را بدهد. حال در این سیستم باید تعداد سنسورها و ویژگی‌های آنها از جمله زمان تشخیص و نیز delay مربوط به درک محیط شبیه‌سازی شود. همچنین سیستم باید در سناریوهای مختلف مورد آزمایش قرار بگیرد؛ مثلاً سناریویی در نظر بگیریم که راننده بیش از اندازه به ماشین رو به روی نزدیک شود و یا اینکه برای سیستم ترمز ایرادی بوجود آید. در نهایت پس از این شبیه‌سازی یک سیستم کنترل حادثه داریم که در کمترین زمان ممکن، بهترین پاسخ ممکن را می‌دهد و بر اساس شبیه‌سازی می‌توانیم زمان مورد نیاز برای سنسورها، تعداد سنسورها و نیز دقت پیشبینی حادثه را بالا ببریم.

سوال 4.

الف) ابتدا جدول ورودی را رسم می‌کنیم تا ببینیم هر تسک در چه زمانی وارد سیستم شده است.

Task Number	Arival Time
1	0
2	3
3	5
4	6
5	8
6	9
7	11
8	12
9	14
10	17

سپس Timestamp را رسم می‌کنیم تا تحلیل کنیم در هر زمانی کدام کار در کدام GPU قرار دارد.

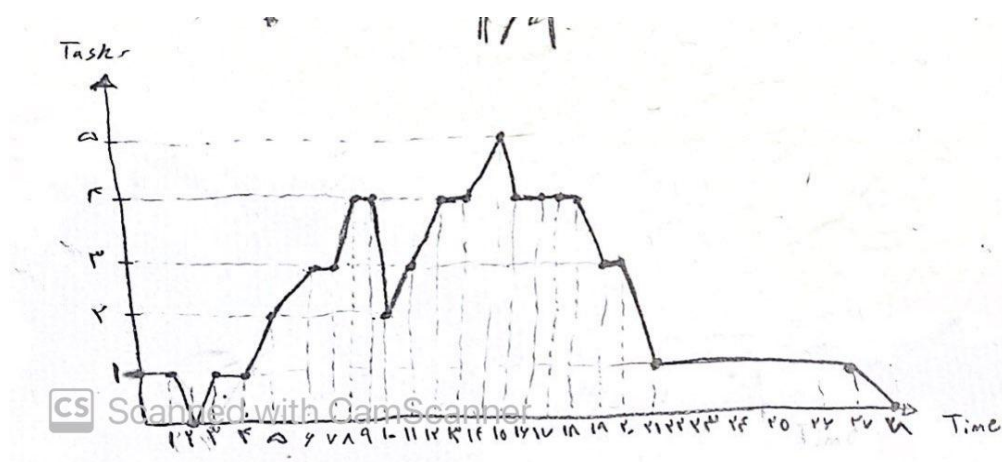
Time Stamp	GPU 1 Task	GPU 2 Task
0	-	1
1	-	1
2	-	-

3	2	-
4	2	-
5	2	3
6	2	3
7	2	3
8	2	3
9	2	4
10	5	6
11	5	6
12	5	6
13	5	6
14	5	6
15	5	7
16	5	7
17	8	7
18	8	7
19	8	9
20	8	9
21	-	7
22	-	7
23	-	7
24	-	7
25	-	7
26	-	7
27	-	7

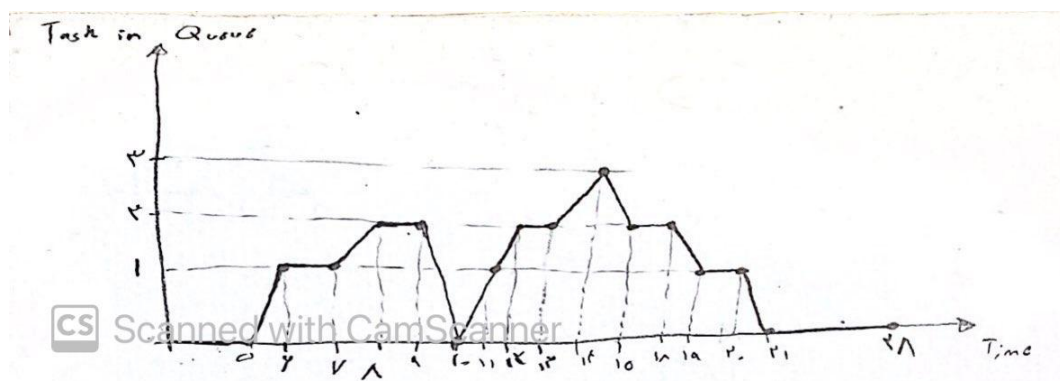
حال جدول کلی را رسم می‌کنیم.

Task Number	Arrival Time	Time Service Begins	GPU Number	Service Time	Time Service Ends	Waiting Time
1	0	0	2	2	2	0
2	3	3	1	7	10	0
3	5	5	2	4	9	0
4	6	9	2	1	10	3
5	8	10	1	7	17	2
6	9	10	2	5	15	1
7	11	15	2	4	19	4
8	12	17	1	4	21	5
9	14	19	2	2	21	5
10	17	21	2	7	28	4

ب) نمودار تسک‌های حاضر در سیستم بر حسب زمان :



پ) می‌دانیم در هر لحظه حداکثر 2 تسک می‌توانند در سیستم در حال انجام باشند پس در نقاطی از نمودار که بیش از 2 تسک دارند صف انتظار تشکیل می‌شود.



(ت)

همانطور که از جدول بالا مشخص است کارت گرافیک اول 10 واحد زمانی بیکار بوده و کارت گرافیک دوم 3 واحد زمانی.

(ث)

در جدول کلی یک ستون waiting time است که مشخص می‌کند هر تسک از زمانی که وارد شده، چقدر طول کشیده تا یکی از GPU ها برروی آن شروع به کار کرده. همانطوری که مشخص است :

$$\text{Avg Waiting time} = (0 + 0 + 0 + 3 + 2 + 1 + 4 + 5 + 5 + 4)/10 = 2.4$$

(ج)

1. می‌توان از تعداد GPU های بیشتری استفاده کرد که طول صف را کاهش می‌دهد.
2. می‌توان

سوال 5.

برای حل این سوال یک کد پایتون کوچک طراحی می‌کنیم که جواب را بهتر نمایش دهیم.

****نوتبوک jupyter این سوال همراه این pdf ضمیمه شده است.****

ابتدا باید ببینیم در هر روز چند مشتری وارد مغازه می‌شوند.

برای این کار از تابع numpy کمک می‌گیریم.

```
np.random.choice(Custom_list, p=Prob_list)
```

این تابع ورودی را به همراه توزیع احتمالی گرفته و داده تصادفی تولید می‌کند.

سپس باید به ازای هر مشتری تعداد نان‌های خریداری شده را نیز دوباره با استفاده از

توزیع داده شده در سوال به صورت تصادفی تولید کنیم.

```
def find_day_bread():
    num_of_customers =
generate_custon(Customer_Num, Customer_Prob)
    sum_bread = 0
    for customer in range(num_of_customers):
        sum_bread += generate_custon(Bread_Bought, Bread_Prob)
    return sum_bread
```

حال جواب تولید شده را بررسی می‌کنیم.

Total breads = [41, 34, 36, 33, 31, 34, 46, 32, 37, 48]

این تعداد کل نان‌های خواسته شده توسط مشتریان در 10 روز است.

حال باید بهترین حالت را پیدا کنیم.

می‌دانیم در صورت فروش هر نان 1200 تومان سود می‌کند و در صورتی که نانی باقی

بماند با -200 تومان ضرر آنرا به فروش می‌رساند.

پس کد زیر را جهت گرفتن بهترین عدد در نظر گرفتیم.

```
best_count = 0
best_money = 0
```

```

for i in range(min(bought_breads), max(bought_breads) + 1):
    money = 0
    for bread_count in bought_breads:
        money += max(i - bread_count, 0) * (-200) +
min(i, bread_count) * (1200)
    if money > best_money:
        best_money = money
        best_count = i
print(best_count)

```

حال جواب داده شده 46 است یعنی اگر هر روز 46 نان بپزد، بیشترین میزان سود را می‌کند.

در این مثال ما تعداد مشتریان و تعداد نان‌هایی خریداری شده را شبیه‌سازی کردیم و نیز در آخر بدست آوردیم که به صورت متوسط اگر هر روز 46 نان بپزد بیشترین سود را خواهد کرد.

Day	Customers	Bread Count	Sum Breads
1	20	[2, 1, 4, 2, 4, 1, 1, 1, 1, 1, 2, 2, 4, 2, 4, 1, 1, 2, 1, 4]	41
2	18	[1, 1, 1, 1, 4, 2, 4, 1, 1, 1, 1, 1, 3, 3, 1, 3, 1, 4]	34
3	20	[1, 4, 1, 1, 1, 4, 3, 3, 1, 1, 1, 1, 1, 3, 1, 3, 3, 1, 1]	36
4	18	[1, 1, 1, 1, 3, 2, 1, 4, 3, 3, 1, 1, 3, 4, 1, 1, 1, 1]	33
5	18	[1, 1, 2, 1, 2, 2, 1, 4, 1, 1, 2, 1, 2, 1, 1, 4, 3, 1]	31
6	16	[1, 1, 4, 1, 4, 1, 1, 4, 4, 1, 4, 1, 1, 1, 4, 1]	34
7	18	[4, 4, 1, 1, 4, 4, 1, 1, 4, 1, 4, 3, 3, 1, 1, 1, 4, 4]	46
8	16	[1, 4, 1, 1, 1, 1, 3, 4, 1, 1, 1, 1, 1, 3, 4, 4]	32
9	16	[2, 4, 4, 3, 2, 4, 1, 1, 4, 1, 2, 3, 1, 3, 1, 1]	37
10	20	[1, 1, 1, 4, 4, 4, 4, 1, 4, 4, 4, 1, 4, 1, 1, 1, 1, 1, 2, 4]	48

حال به روش دیگری نیز دوباره شبیه‌سازی را انجام می‌دهیم.
 دو جدول برای مشتری‌ها و نان‌های خریده رسم می‌کنیم.

prob	cumulative	random number	customers
0.2	0.2	1 - 20	14
0.28	0.48	21 - 48	16
0.3	0.78	49 - 78	18
0.22	1	78 - 100	20

prob	cumulative	random number	bread
0.5	0.5	1 - 50	1
0.15	0.65	51 - 65	2
0.1	0.75	66 - 75	3
0.25	1	76 - 100	4

حال صرفاً اعداد رندوم تولید کرده و می‌بینیم در کدام بازی قرار می‌گیرند و بر طبق آنان شبیه‌سازی را انجام می‌دهیم.

[.91, .83, .81, .83, .26, .73, .80, .31, .76, .85]