CS-350 - Fundamentals of Computing Systems
Homework Assignment #4

Due on October 20, 2020 at 11:59 pm

Prof. Renato Mancuso

Renato Mancuso

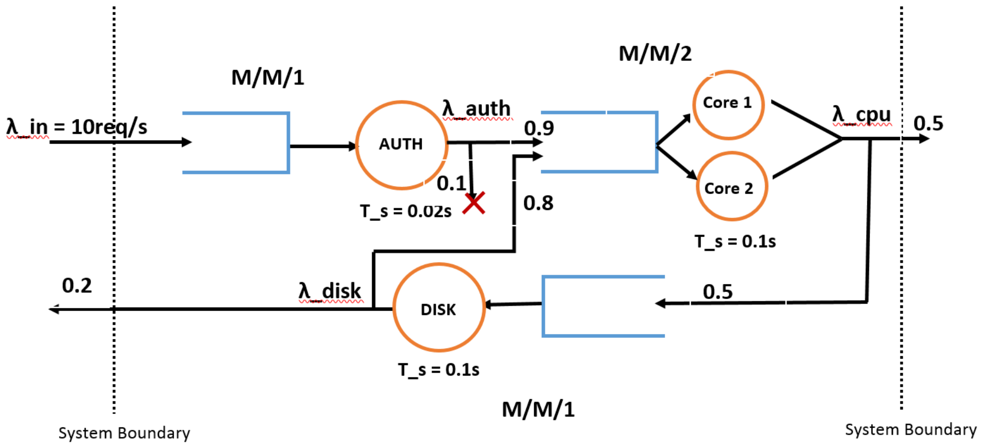


Figure 1: Queueing diagram for Q1.

Problem 1

An online transaction processing (OLTP) system is an ensemble of three key components. The first component is the authentication module (AUT) where are authorization of all incoming requests are verified. The service time for the AUT follows an exponential distribution with a mean of 20 ms, and the probability that an incoming request is authentic is 0.9. The service is considered to be terminated for all unauthorized requests. All authorized requests are then forwarded to the CPU for processing, which serves requests on two cores in parallel and with a global queue. The CPU service time is distributed exponentially with a mean of 100 ms. After further processing at the CPU, a service completes and leaves the system with 0.5 probability. The remaining requests further need to access the disk (HDD), the service time of which is known to be exponential with a mean of 100 ms. A request served by the HDD, requires further processing within the CPU with 0.8 probability. For the remaining requests, the service is considered to be complete. You know that the arrival of requests in the OLTP system follows a Poisson distribution with a mean of 10 requests per second. Answer the following.

a) Draw the queuing diagram for the above OLTP system.

Ans-1a Refer to Figure 1.

b) What is the arrival rate of requests at the CPU?

Ans-1b In steady-state $\lambda_{in} = \lambda_{out}$. This implies

$$\begin{aligned} \lambda_{AUTH} &= 10 = \lambda_{in} \\ \lambda_{CPU} &= 0.9\lambda_{AUTH} + 0.8\lambda_{DISK} \\ \lambda_{DISK} &= 0.5\lambda_{CPU} \end{aligned} \tag{1}$$

Solving the above equations gives us $\lambda_{CPU} = 15$ req/s and $\lambda_{DISK} = 7.5$req/s

c) Among the AUT, CPU, and HDD, which one is the bottleneck of the system and why?

Ans-1c Calculating Utilizations for each subsystem:

$$\begin{aligned} \rho_{CPU} &= \frac{\lambda_{CPU} * T_{s_{CPU}}}{2} = 0.75 \\ \rho_{AUTH} &= \lambda_{AUTH} * T_{s_{AUTH}} = 0.2 \\ \rho_{DISK} &= \lambda_{DISK} * T_{s_{DISK}} = 0.75 \end{aligned} \tag{2}$$

Thus both CPU and DISK are bottlenecks of the system as they have the highest utilization.

d) What is the average number of requests present in the system at any point in time?

Ans-1d Using Jackson's Theorem: $q_{total} = q_{CPU} + q_{AUTH} + q_{DISK}$. Since AUTH and DISK are M/M/1 systems thus:

$$q_{\{AUTH,DISK\}} = \frac{\rho_{\{AUTH,DISK\}}}{1 - \rho_{\{AUTH,DISK\}}} \tag{3}$$

Thus $q_{AUTH} = 0.25$ requests and $q_{DISK} = 3$ requests. For CPU (M/M/2) system:

$$\begin{aligned} N &= 2 \\ C &= \frac{2 * \rho_{CPU}^2}{1 + \rho_{CPU}} = 0.643 \\ q_{CPU} &= 2\rho_{CPU} + C\frac{\rho_{CPU}}{1 - \rho CPU} = 3.43req \end{aligned} \tag{4}$$

Thus $q_{total} = 6.68$ requests.

e) What is the overall slowdown that requests are subject to, because of queuing effects in the system?

Ans-1e We know that $slowdown = \frac{meanresponsetime}{minresponsetime}$. Now $T_q = q_{total}/\lambda_{in} = 6.68/10 = 0.668$s. For min response time we set $\lambda = 0.001$req/s i.e. a very small value. Recalculating λs for CPU, AUTH and DISK in a similar manner as part-1b yields: $\lambda_{AUTH} = 0.001$req/s , $\lambda_{CPU} = 1.5 * 10^{-3}$req/s and $\lambda_{DISK} = 7.5 * 10^{-4}$req/s. Similarly Utilizations for each subsystem become : $\rho_{CPU} = 7.5 * 10^{-5}, \rho_{AUTH} = 2*10^{-5}, \rho_{DISK} = 7.5*10^{-5}$. Using Jackson's theorem to calculate q for each subsystem gives: $q_{AUTH} = 2*10^{-5}, q_{DISK} = 7.5*10^{-5} and q_{CPU} = 1.5*10^{-4}$ requests. $q_{total} = 2.45*10^{-4}$requests. Thus $T_{q_{min}} = 2.45 * 10^{-4}/0.001 = 0.245$s. Therefore slowdown = 0.668/0.245 = 2.73.

f) Would you be able to solve the system described in this problem with the same approach you used for Part a)-e) if your were told that the queue at the HDD had a limited size? Explain why or why not. Regardless, briefly mention what could be an alternative approach.

Ans-1f No, with a limited size queue at the DISK subsystem, we can no longer apply Jackson's theorem to solve the system as the arrival of events to the DISK subsystem are no longer Poisson due to the internal looping of requests. An alternative to the analytical approach is via Discrete Event Simulation.

Table 1: Performance Measurements for PoCs 1 to 3

Sample #	PoC 1	PoC 2	PoC 3
1	68	75	76
2	37	112	81
3	53	106	81
4	70	52	78
5	47	109	76
6	44	106	74
7	58	76	78
8	42	118	70
9	44	76	73
10	45	92	79
11	52	98	73
12	50	21	81
13	61	27	81
14	40	112	74
15	44	68	72
16	54	120	84
17	34	86	73
18	69	29	79
19	58	66	81
20	72	27	72
21	62	93	72

Problem 2

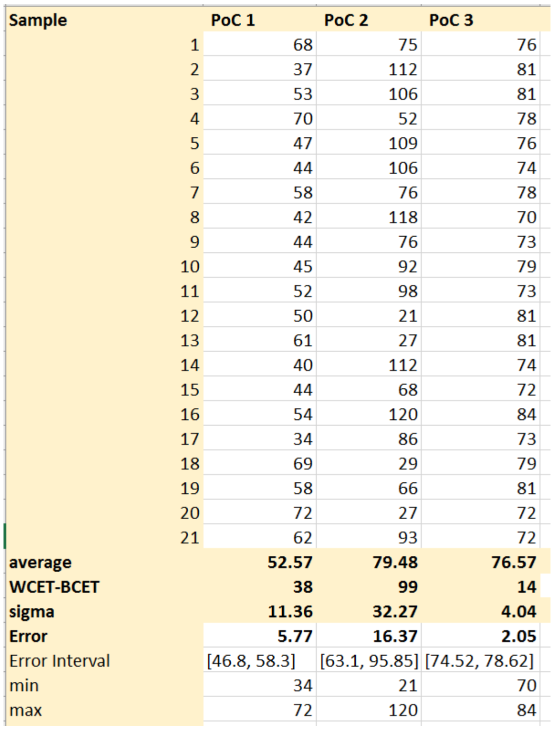
You are the head of the customers contracts division at Solutions for Cloud with Awesome Machines, or SCAM for short. The business model of SCAM. is the following: they buy some antiquated machines from Craigslist, measure how well they work, and decide which kind of performance guarantee they can sell to customers. A performance guarantee, also called a Service Level Agreement (SLA) is a legally binding statement of the type: "I guarantee that your application will take between a and b milliseconds to run, in at least x cases out of y".
Assume that all your customer want to run the same application, and consider the runtime measurements that your intern performed on a fresh batch of Purchased-off-Craigslist machines, or PoC, for short. The measurements for three of these PoC's are reported in Table 1.
Provide a report to the CEO of SCAM with the following considerations:

a) What PoC is more suitable for customers that just want better performance on average. Motivate your answer.

Ans-2a Refer to Figure 2: POC 1 has better performance on average since it has minimum average time.

b) What machine is preferable for customers who want a more deterministic behavior, in spite of worse overall performance. Motivate your answer.

Ans-2b Refer to Figure 2: POC 3 is more deterministic as it has the least worst case variation between the worst-case and best-case runtime as well as minimum standard deviation.



Sample	PoC 1	PoC 2	PoC 3
1	68	75	76
2	37	112	81
3	53	106	81
4	70	52	78
5	47	109	76
6	44	106	74
7	58	76	78
8	42	118	70
9	44	76	73
10	45	92	79
11	52	98	73
12	50	21	81
13	61	27	81
14	40	112	74
15	44	68	72
16	54	120	84
17	34	86	73
18	69	29	79
19	58	66	81
20	72	27	72
21	62	93	72
average	**52.57**	**79.48**	**76.57**
WCET-BCET	**38**	**99**	**14**
sigma	**11.36**	**32.27**	**4.04**
Error	**5.77**	**16.37**	**2.05**
Error Interval	[46.8, 58.3]	[63.1, 95.85]	[74.52, 78.62]
min	34	21	70
max	72	120	84

Figure 2: Calculations for Problem 2.

c) Can we offer a SLA with $a = 45\ ms$, $b = 60\ ms$, $x = 980$, $y = 1000$? If so, which PoC should be assigned to users that sign a contract with this SLA?

Ans-2c Required interval: [45 60] and confidence $= x/y = 980/1000 = 0.98 \implies 98\%$
$1 - \alpha = 0.98 \implies \alpha = 0.02 \implies 1 - \alpha/2 = 0.99$
$P(Z_{\alpha/2}) = 2.325$ from the lookup table. Refer to Figure 2 for confidence error intervals of all 3 POCs for the required 98% confidence. Since POC1's interval is closest to the required interval of[45 60] thus POC1 should be assigned to users signing a contract with this SLA.

d) Can we offer a SLA on PoC 2, where $x = 999$ and $y = 1000$? If so, what should we pick for a and b to prevent being sued?

Ans-2d $x/y = 0.999 = 1 - \alpha \implies 1 - \alpha/2 = 0.9995$. From the lookup table $Z_{\alpha/2} = 3.295$. Thus $E = \frac{(Z_{\alpha/2})*(\sigma_{POC2})}{\sqrt{21}} = 23.20$. This means the interval $[a, b] = [56.28, 102.7]$. Since this interval falls between the min and max runtime values for POC2 (refer to table above) hence we can offer an SLA on POC2 with 99.9% confidence.

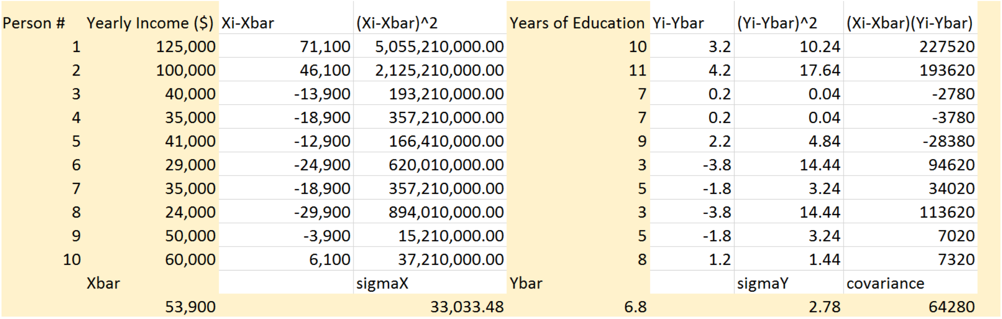
e) A customer has offered to pay way more than the regular folks as long as we can guarantee a SLA with $a = 75$, $b = 78$ and $x = 98$, $y = 100$. Can we offer this SLA right away with PoC 3? If not, how many more samples you should tell your intern to collect?

Ans-2d Required Confidence $= 1 - \alpha = x/y = 98/100 = 0.98 \implies 1 - \alpha/2 = 0.99$ From the Lookup table we see that $Z_{\alpha/2} = 2.325$. Using N=21 samples the Error becomes:

$E = 2.325 * 4.04/\sqrt{21} = 2.05$. We know that the required minimum error is : $E_{required} = min(76.57 - 75, 78 - 76.57) = min(1.57, 1.43) = 1.43$ thus more samples are needed to reduce the error to 1.43. $\implies \sqrt{N} = 2.325 * 4.04/1.43 \implies N = 43.15 \approx 44$. Thus 44-21 = 23 more samples are needed to be collected.

Person #	Yearly Income ($)	Years of Education
1	125000	10
2	100000	11
3	40000	7
4	35000	7
5	41000	9
6	29000	3
7	35000	5
8	24000	3
9	50000	5
10	60000	8

Table 2: Incomes and Years of Education for 10 individuals in CS.



Person #	Yearly Income ($)	Xi-Xbar	(Xi-Xbar)^2	Years of Education	Yi-Ybar	(Yi-Ybar)^2	(Xi-Xbar)(Yi-Ybar)
1	125,000	71,100	5,055,210,000.00	10	3.2	10.24	227520
2	100,000	46,100	2,125,210,000.00	11	4.2	17.64	193620
3	40,000	-13,900	193,210,000.00	7	0.2	0.04	-2780
4	35,000	-18,900	357,210,000.00	7	0.2	0.04	-3780
5	41,000	-12,900	166,410,000.00	9	2.2	4.84	-28380
6	29,000	-24,900	620,010,000.00	3	-3.8	14.44	94620
7	35,000	-18,900	357,210,000.00	5	-1.8	3.24	34020
8	24,000	-29,900	894,010,000.00	3	-3.8	14.44	113620
9	50,000	-3,900	15,210,000.00	5	-1.8	3.24	7020
10	60,000	6,100	37,210,000.00	8	1.2	1.44	7320
	Xbar		sigmaX	Ybar		sigmaY	covariance
	53,900		33,033.48	6.8		2.78	64280

Figure 3: Calculations for Problem 3.

Problem 3

You have just been elected to be the next chair of the BU Computer Science Department. You are now trying to understand what is the average income that your students should expect when completing their degrees at various stages of education (e.g. BS, MS, PhD, Postdoc, etc.). You have sampled 10 individuals pursuing a career in CS at random and obtained the values in Table 2. Now you are wondering about the following.

a) What is the average yearly salary for a person pursuing a career in CS?

Ans-3a Average Yearly Salary $= \frac{125k+100k+40k+35k+41k+29k+35k+24k+50k+60k}{10} = 53900$ $

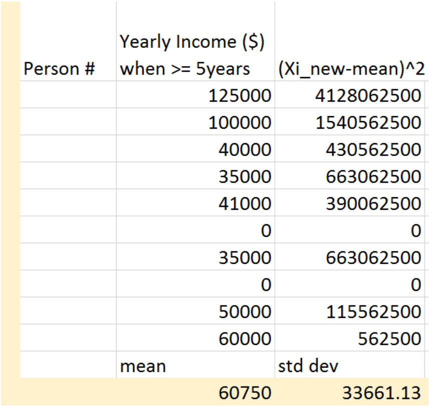
b) What confidence do you have that the average salary you computed is accurate within a ±$10000 range?

Ans-3b We know E=10000. Standard Deviation of the sample: $\sigma_X = 33033.48$ $. $Z_{\alpha/2} = \frac{E}{\frac{\sigma_X}{\sqrt{N}}} = \frac{10000}{\frac{33033.48}{\sqrt{10}}} = 0.96$. Looking at the look-up table for standard normal distribution, we find that $P(Z_{\alpha/2}) = P(0.96) = 0.8315 = 1 - \frac{\alpha}{2}$. Thus $\alpha = 0.337 \implies$ a confidence of 66.3%.

c) How many additional people you should interview to increase the confidence on your estimate to 95%?

Ans-3c A confidence of 95% $\implies \alpha = 1 - 0.95 = 0.05$. It follows that $1 - \frac{\alpha}{2} = 0.975$.Now we look at the Std. Distribution look-up table and find the value $Z_{\alpha/2}$ for which $F(Z_{\alpha/2}) = 0.975$. This results in a value of $Z_{\alpha/2} = 1.96$. For the same E=10000 and σ_X from previous answer and using $Z_{\alpha/2} = \frac{E}{\frac{\sigma_X}{\sqrt{N}}}$,

Problem 3 continued on next page. . .



Person #	Yearly Income ($) when >= 5years	(Xi_new-mean)^2
	125000	4128062500
	100000	1540562500
	40000	430562500
	35000	663062500
	41000	390062500
	0	0
	35000	663062500
	0	0
	50000	115562500
	60000	562500
	mean	std dev
	60750	33661.13

Figure 4: Calculations for 3e.

we get $N = (\frac{\sigma_X \times Z_{\alpha/2}}{E})^2 = 41.92 \approx 42$. Additional people needed to interview = 42 - 10 = 32 people with the same confidence interval and estimate of mean and standard deviation as in part 3a and 3b.

d) Can you conclude that there is a relation between a longer education in CS and the received yearly salary?

Ans-3d We calculate correlation of the two sample sequences in Table: 3. Standard Deviation for years of Education = $\sigma_Y = 2.78$. $Corr(X,Y) = \frac{Cov(X,Y)}{\sigma_X \times \sigma_Y} = \frac{1}{10} \frac{\sum_{i=1}^{10}(X-\overline{X})\times(Y-\overline{Y})}{\sigma_X \times \sigma_Y} = 0.699$. Since the correlation is positive we conclude that a longer education in CS is related to received yearly salary.

e) If you decide to "stay in school" for 5 years or more, what confidence you have that you will be able to have a yearly salary that is somewhere between \$ 50000 and \$ 71500?

Ans-3e Number of people who stay for 5 years or more : N = 8. New Mean for Years of Education = 7.75 years and New Mean for Income = 60750\$. Std Dev of income = 33661.13\$ Now E= $60750 - 50000 = 10750$. Therefore $Z_{\alpha/2} = 10750 \times \sqrt{8}/33661.13 = 0.903$. From the lookup table $P(Z_{\alpha/2}) = 1 - (\alpha/2) = 0.8159 \implies \alpha = 0.3682$ and $1 - \alpha = 0.632 = 63.2\%$

Problem 4

Code integration: In this problem you will develop code to simulate an entire system with multiple servers. The following is the workflow of the system. First, requests arrive with rate λ and enter at server $S0$ which is a single-processor system with average service time T_{s0}. From here, the request goes to either $S1$ or $S2$ with probability $p_{0,1}$ and $p_{0,2}$, respectively. $S1$ has a single infinite queue and **two** processors, each with average service time T_{s1}. $S2$ has a single processor and K_2 maximum queue size (this includes the request being currently served). The processor can serve a request in T_{s2} time. Any request that completes processing at $S1$ or $S2$ always goes to $S3$. This is a single-processor system with service time following a distribution whose PMF is given via 6 parameters $t_1, p_1, t_2, p_2, t_3, p_3$ where t_i is the time it takes to process a request, and p_i is the probability that it will take t_i time for a request to be processed at $S3$. After a request completes at $S3$, it is released from the system with probability $p_{3,out}$, it goes back to S_1 with probability $p_{3,1}$, or it goes back to $S2$ with probability $p_{3,2}$. Assume all service times for $S0 - S2$, as well as inter-arrival times of requests from the outside at $S0$ are exponentially distributed.
Write a Java class namely "Simulator" in its own file `Simulator.java` that implements a discrete event simulator. The simulator should have a method with the following prototype: `void simulate(double time)`, that simulates the arrival and execution of requests at the system described above for `time` milliseconds, where `time` is passed as a parameter to the method. You should re-use code you wrote as part of the previous assignments.
Apart from implementing the `simulate(...)` method, the class should also include a `public static void main(String [] args)` function. The `main(...)` function should accept 17 parameters from the calling environment (in the following order):

1. length of simulation time in milliseconds. This should be passed directly as the `time` parameter to the `simulate(...)` function.
2. average arrival rate of requests at the system λ;
3. average service time T_{s0} at $S0$;
4. average service time T_{s1} at $S1$;
5. average service time T_{s2} at $S2$;
6. service time t_1 at $S3$;
7. probability p_1 of service time t_1 at $S3$;
8. service time t_2 at $S3$;
9. probability p_2 of service time t_2 at $S3$;
10. service time t_3 at $S3$;
11. probability p_3 of service time t_3 at $S3$;
12. K_2 maximum length of the queue expressed in number of requests at $S2$;
13. routing probability $p_{0,1}$ that a request will go from $S0$ to $S1$;
14. routing probability $p_{0,2}$ that a request will go from $S0$ to $S2$;
15. routing probability $p_{3,out}$ that a request will exit the system from $S3$;
16. routing probability $p_{3,1}$ that a request will go from $S3$ back to $S1$;
17. routing probability $p_{3,2}$ that a request will go from $S3$ back to $S2$;

Problem 4 continued on next page. . .

All times should be intended in milliseconds.
It is responsibility of the `main(...)` function to internally invoke the implemented `simulate(...)` function **only once**. The `simulate(...)` function will need to print in the console the simulated time at which each request arrives at the system (`ARR`), initiates service at Si (`START Si`), and completes service (`DONE Si`). when a new request arrives at $S2$ while the current total number of requests in the system is already K_2 ($S2$ is full), then the newly arrived request is dropped and discarded (see `DROP S2`). The output must look like this:

```
R0 ARR: <timestamp>
R0 START S0: <timestamp>
R1 ARR: <timestamp>
R2 ARR: <timestamp>
R0 DONE S0: <timestamp>
R0 FROM S0 TO S1: <timestamp>
R0 START S1,1: <timestamp>
R1 START S0: <timestamp>
R1 DONE S0: <timestamp>
R1 FROM S0 TO S2: <timestamp>
R1 START S2: <timestamp>
R2 START S0: <timestamp>
R2 DONE S0: <timestamp>
R2 FROM S0 TO S2: <timestamp>
R3 ARR: <timestamp>
R3 START S0: <timestamp>
R3 DONE S0: <timestamp>
R3 FROM S0 TO S2: <timestamp>
R4 ARR: <timestamp>
R4 START S0: <timestamp>
R4 DONE S0: <timestamp>
R4 FROM S0 TO S2: <timestamp>
R4 DROP S2: <timestamp>
R5 ARR: <timestamp>
R5 START S0: <timestamp>
R6 ARR: <timestamp>
R7 ARR: <timestamp>
R5 DONE S0: <timestamp>
R5 FROM S0 TO S1: <timestamp>
R5 START S1,2: <timestamp>
R6 START S0: <timestamp>
R0 DONE S1,1: <timestamp>
R0 FROM S1 TO S3: <timestamp>
R0 START S3: <timestamp>
R0 DONE S3: <timestamp>
R0 FROM S3 TO OUT: <timestamp>

S0 UTIL: <utilization of S0>
S0 QLEN: <avg. number of requests in S0>
S0 TRESP: <avg. response time of requests at S0>

S1,1 UTIL: <utilization of S1 processor 1>
```

```
S1,2 UTIL: <utilization of S1 processor 2>
S1 QLEN: <avg. number of requests in S1>
S1 TRESP: <avg. response time of requests at S1>

S2 UTIL: <utilization of S2>
S2 QLEN: <avg. number of requests in S2>
S2 TRESP: <avg. response time of requests at S2>
S2 DROPPED: <total number of dropped requests>

S3 UTIL: <utilization of S3>
S3 QLEN: <avg. number of requests in S3>
S3 TRESP: <avg. response time of requests at S3>

QTOT: <avg. number of requests in the entire system>
TRESP: <avg. response time of the entire system>
```

where `<timestamp>` is the simulated time in milliseconds at which the event occurred printed in decimal format. This is also called the **trace** of the simulation. Also, `<utilization>` is a decimal number between 0 and 1, `<avg. queue lenght>` is a decimal number, and `<avg. response time of requests>` is a decimal number expressed in milliseconds.

Submission Instructions: in order to submit this homework, please follow the instructions below for exercises and code.

The solutions for Problem 1-3 should be provided in PDF format, placed inside a single PDF file named `hw4`.pdf and submitted via Gradescope. Follow the instructions on the class syllabus if you have not received an invitation to join the Gradescope page for this class. You can perform a partial submission before the deadline, and a second late submission before the late submission deadline.

The solution for Problem 4 should be provided in the form of Java source code. To submit your code, place all the `.java` files inside a compressed folder named `hw4`.zip. Make sure they compile and run correctly according to the provided instructions. The first round of grading will be done by running your code. Use CodeBuddy to submit the entire `hw4`.zip archive at `https://cs-people.bu.edu/rmancuso/courses/cs350-fa20/scores.php?hw=hw4`. You can submit your homework multiple times until the deadline. Only your most recently updated version will be graded. You will be given instructions on Piazza on how to interpret the feedback on the correctness of your code before the deadline.