Quiz 2

Instructions:

- Do not open this quiz booklet until directed to do so. Read all the instructions on this page.
- Write your name below and circle your recitation at the bottom of this page.
- Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem. Pages will be separated for grading.
- Make sure you write your first name on **every** page of the exam after the first page.
- **You are allowed two double-sided, letter-sized sheets with your own notes**. No calculators or programmable devices are permitted. No cell phones or other communication devices are permitted.
- Refrain from discussing this exam until Wednesday, April 19.

Advice:

- You have 120 minutes to earn a maximum of 120 points. **Do not spend too much time on any single problem.** Read them all first, and attack them in the order that allows you to make the most progress.
- When writing an algorithm, a **clear** description in English will suffice. Using pseudo-code is not required.
- Do not waste time re-deriving facts that we have studied. Simply state and cite them.

Question	Parts	Points
1: True or False	6	18
2: Random Walks	1	10
3: Hashing	3	19
4: Spy Game!	4	34
5: Triangle Counting	3	23
6: Expected Utility of Nash Equilibrium	2	16
Total:		120

Name: ______________________________

Circle your recitation:	R01	R02	R03	R04	R05	R06	R07	R08	R09
	Emanuele Ceccarelli	Shraman Chaudhuri	Mayuri Sridhar	Kai Xiao	Varun Mohan	Isaac Grosof	Devin Neal	Lei “Jerry” Ding	Sagar Indurkhya
	10 AM	**11 AM**	**12 PM**	**1 PM**	**2 PM**	**3 PM**	**11 AM**	**12 PM**	**1 PM**

Problem 1. [18 points] **True or False** (6 parts)
Please circle **T** or **F** for the following. *No justification is needed.*

(a) [3 points] **T F** A (non-lazy) random walk converges to a stationary distribution on every undirected tree graph.

> **Solution:** False. Trees are bipartite graphs.

(b) [3 points] **T F** Given a Las Vegas algorithm for a particular problem, one can construct a Monte Carlo algorithm for the same problem by running the Las Vegas algorithm for a certain amount of time and returning (i) the Las Vegas solution if it completes, or (ii) a randomly selected solution otherwise.

> **Solution:** True. If you wait long enough, e.g. three times the expected runtime, this works.

(c) [3 points] **T F** Consider a family of hash functions $\mathcal{H} = \{h_1, h_2\}$ that maps a universe $\{a, b, c\}$ to $\{0, 1\}$, where $h_1(a) = h_1(b) = h_1(c) = 0$ and $h_2(a) = h_2(b) = h_2(c) = 1$. This family $\mathcal{H}$ gives rise to a universal hash function family.

> **Solution:** False. E.g., elements a and b collide in both h_1 and h_2.

(d) [3 points] **T F** The two-person game with the following payoff matrix has a (mixed) Nash equilibrium:

$$\begin{bmatrix} (1,-1) & (3,-3) & (-1,1) \\ (2,-2) & (-1,1) & (2,-2) \\ (-4,4) & (0,0) & (1,-1) \end{bmatrix}$$

> **Solution:** True. It is a zero-sum game, so, by the Min–Max Theorem, it has to have a Nash equilibrium.

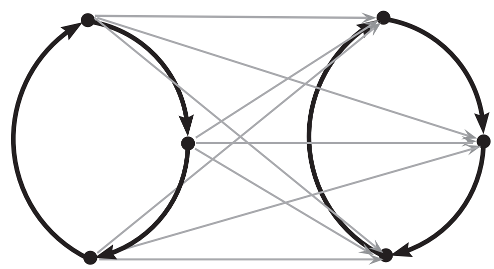
(e) [3 points] **T F** The analysis of Las Vegas algorithms bounds the expected running time of the algorithm over all possible inputs.

> **Solution:** Because this question could be interpreted multiple ways, we accepted either True or False here.

(f) [3 points] **T F** If v is a vertex sampled from the stationary distribution of a connected undirected graph, and u is the vertex visited immediately after taking one step of the (non-lazy) random walk then (u, v) is a uniformly random edge of the graph.

> **Solution:** True.

Problem 2. [10 points] **Random Walks** (1 part)
Consider the graph depicted below. It consists of two copies of a directed cycle of length 3 and also has edges going from *each* vertex of the "left" copy to *each* vertex of the "right" copy. (So, there is a total of 9 of such "left" to "right" edges.)



Describe a stationary distribution of this graph.

Solution:

Pre-Solution:

Observe that in this graph, once we leave the left cycle, there is no way of getting back to it. This means that the stationary distribution π has to be supported only on the right cycle, i.e., $\pi_{v_i^L} = 0$ for all i.

Solution:

We have to have that

$$\pi_{v_{i+1}^R} = \pi_{v_i^R},$$

for all i (interpreting v_4^R as v_1^R). So, as the right cycle has exactly 3 vertices and π has to be a normalized distribution, we can conclude that

$$\pi_v = \begin{cases} 0 & \text{if } v = v_i^L, \text{ for some } i \\ \frac{1}{3} & \text{otherwise, i.e., if } v = v_i^R, \text{ for some } i. \end{cases}$$

Post-Solution:

Although a directed graph must be strongly connected and aperiodic for it to converge to a stationary distribution, the existence of a stationary distribution only requires that we be able to apply the walk matrix to a distribution and it doesn't change. Here we are asked only to find a stationary distribution, and not to find one to which one would necessary converge.

Problem 3. [19 points] **Hashing** (3 parts)
In perfect hashing, we had a hash-table which used multiple smaller hash-tables to resolve collisions in the former. Here, we look at what would happen if we just had two hash-tables of the same size instead. Consider the following method of producing a hash function with $O(1)$ collisions. Given n items, we allocate two arrays of size $n^{1.5}$. When inserting a new item, we map it to one slot in each array and place it in the emptier of the two slots (that is, if both are full, it is assigned to the slot with the shorter chain).

Assume a uniformly random function is used to map an item to a slot, and that we use chaining if we get collisions even in the "emptier" slot. Also recall that $\sum_{x=1}^{n} x^2 = n(n+1)(2n+1)/6$.

(a) [5 points] Show that the the probability that item $k+1$ collides with any of the previously inserted items is $\leq k^2/n^3$.

Solution:

Pre-Solution:

For item k+1 to collide, the item must hash into a slot that is not empty. For this to happen in our hashing scheme, we can define the two slots that item k+1 can hash into as $h_1(k+1)$ and $h_2(k+1)$, which represent the slot of item $k+1$ in table 1 and table 2, respectively.

Solution:

If item $k+1$ collides, then $h_1(k+1)$ and $h_2(k+1)$ are both occupied. We note that these are not independent events - however, the probability that $h_2(k+1)$ is occupied actually decreases if we know that $h_1(k+1)$ is occupied - so assuming the events are independent will actually give us an upper bound on the probability. We know that each table contains at most k items; thus, the probability that a collision will occur in table 1 will be at most $\frac{k}{n^{1.5}}$. The same logic can be used to calculate the probability of a collision in table 2. From here, we can assume the events are independent to find the probability that item $k+1$ collides. That gives us the following bound:

$$\Pr[k+1 \text{ collides}] \leq (\frac{k}{n^{1.5}})^2 = k^2/n^3$$

(b) [7 points] Given that we have n items, use the bound from (a) to show that the expected number of collisions is $\leq \frac{1}{3}$.

Solution:

Pre-Solution:

We can define a random variable X to represent the total number of collisions. There's no quick way to calculate X directly. However, we can see that X only increases when some particular item k collides. Thus, we can define indicator variables X_i which are 1 if item i collides and 0 otherwise. Then, we know that $X = \sum X_i$. From here, when we see the sum, we realize that we can use linearity of expectation, since we know the expected value of X_i from part (a) (recall how the expectation of an indicator variable is equal to the event probability).

Solution:

From part (a), we know that the probability that item k collides with any of the previously inserted items is at most k^2/n^3. Thus, the expected number of times an item is inserted into a bucket which is already occupied is:

$$\sum_{k=0}^{n-1} \frac{k^2}{n^3} = \frac{(n-1)(n)(2n-1)}{6n^3} \leq \frac{1}{3}$$

(c) [7 points] Using the previous parts and Markov's Inequality, give an upper bound on the probability that at least 10 collisions occur.

Solution:

Pre-Solution:

Even if we haven't solved the previous parts, we know from part (b) that the expected number of collisions is at most $1/3$. Given the value of 10 collisions, we can directly plug this into Markov's Inequality to solve for the probability. We will define a random variable equal to the number of collsions. Then, Markov's inequality only requires the expected value to provide any bound.

Solution:

From part (b), we know the expected number of items that collide is at most $\frac{1}{3}$. Thus by Markov's Inequality, we get:

$$\Pr[\text{more than 10 collisions}] \leq \frac{1/3}{10} = \frac{1}{30}$$

Problem 4. [34 points] **Spy Game!** (4 parts)
You have organized a special Spy Meet between the CIA, the MI6, and the SVR, inviting n agents from each organization, for a total of $3n$ agents. (Somewhat unrealistically and boringly, we assume that none of them is a double agent.) None of the agents identifies which organization he/she is working for. (Secrecy is the mark of the trade, after all!)

Still, being a true spy master (and MIT student), you can't resist trying to figure out this information anyway.

The only two pieces of information about their behavior that could help you in this task are:

1. Each agent talks only to people from one of the other agencies, so a CIA agent will never speak to another CIA agent, etc.
2. Each agent speaks with $\geq 3n/4$ agents from each of the other agencies. That is, every CIA agent speaks with at least $3n/4$ of the n MI6 agents, and also to at least $3n/4$ of the n SVR agents (and the same applies to each MI6 and SVR agent).

You have recorded the entire meet, and have a list of all conversations. Given any pair $\{a_1, a_2\}$ of agents, you can query this list to check whether they spoke during the meet in $O(1)$ time. (Note that conversations are symmetric, *i.e.*, $\{a_1, a_2\}$ is the same as $\{a_2, a_1\}$.)

It is not too hard to convince yourself that such a list of all conversations has enough information to figure out the three different groups of agents that each correspond to an agency. In what follows, however, your task will be to come up with a method that not only achieves this task but also does it using as small number of list queries as possible.

(a) [7 points] Given a pair of agents $\{a_1, a_2\}$ from the *same* agency, we say that an agent a is *good for* $\{a_1, a_2\}$ iff a has spoken with *both* a_1 and a_2. Note that, by the first rule of agent behavior, a has to necessarily be from a different agency than the one a_1 and a_2 are from.

Let us fix $\{a_1, a_2\}$ as above and an agency that is different to the agency a_1 and a_2 belong to. Show that there is always at least $\frac{n}{2}$ different agents in that agency that are good for $\{a_1, a_2\}$.
Hint: Rule 2 above might be especially helpful here.

Solution: Without loss of generality, suppose that both a_1 and a_2 are from the CIA. Since both a_1 and a_2 have spoken with at least $\frac{3n}{4}$ different MI6 agents, at least $2 \cdot \frac{3n}{4} - n = \frac{n}{2}$ different MI6 agents have spoken with both of them. Therefore, there are at least $\frac{n}{2}$ good MI6 agents. For the same reason, there are also at least $\frac{n}{2}$ good SVR agents.

(b) [7 points] Let us, again, fix a pair $\{a_1, a_2\}$ of agents from the *same* agency. Given two different agents a and a' that are good for $\{a_1, a_2\}$, we say that a and a' are *special* iff they spoke to each other. (Note that this means that a and a' are from different agencies as well.) Show that for a fixed pair of agents $\{a_1, a_2\}$ as above, there are at least $\frac{n^2}{8}$ such special pairs.

Solution:

Pre-Solution:

In order to count the number of special pairs (a, a') we must first start by knowing that a and a' are from different agencies and they've spoken to a_1 and a_2. Part (a) tells us how many of each we have. Now we need to lower bound how many have talked to each other. The only thing we know here is rule 2. So we should focus on a given a and try to lower bound the number of a's it's talked to.

Solution:

As before, we assume without loss of generality that a_1 and a_2 are from the CIA.

Each MI6 agent that was good for $\{a_1, a_2\}$ has spoken with at least $\frac{3n}{4}$ different SVR agents, $\frac{n}{4}$ of which must be good since, by part (a), there is at least $\frac{n}{2}$ good SVR agents for $\{a_1, a_2\}$.

So, each one of the at least $\frac{n}{2}$ MI6 agent that is good for $\{a_1, a_2\}$ has spoken with at least $\frac{n}{4}$ SVR agents that are also good for $\{a_1, a_2\}$. Each such pair is special. Therefore, there is at least $\frac{n}{2} \cdot \frac{n}{4} = \frac{n^2}{8}$ special pairs.

Post-Solution:

Many students incorrectly translated this problem as having to do with probabilities. This is a deterministic lower bound we are giving, there is so far nothing random happening.

(c) [8 points] Once more, we fix a pair $\{a_1, a_2\}$ of agents from the same agency. Consider choosing two agents a and a', independently at random, from the pool of remaining agents.

Show that the probability that these two sampled agents are special is at least $\frac{1}{36}$.

Solution:

By part (b), we know that there is at least $\frac{n^2}{8}$ of special pairs. As there is at most $\binom{3n-2}{2}$ different pairs of remaining agents. Consequently, the probability that the random pair we chose is special is at least

$$\frac{\frac{n^2}{8}}{\binom{3n-2}{2}} = \frac{n^2}{8} \cdot \frac{2!(3n-4)!}{(3n-2)!} = \frac{n^2}{8} \cdot \frac{2}{(3n-2)(3n-3)} > \frac{n^2}{4(3n)^2} = \frac{1}{36},$$

as desired.

(d) [12 points] Armed with the knowledge from the previous parts, design a Monte Carlo algorithm which, given a pair of agents $\{a_1, a_2\}$ which *might or might not be from the same agency*, outputs "SAME" if a_1 and a_2 belong to the same agency, and outputs "DIFFERENT" if they do not.

Your algorithm should be correct with at least constant probability $c > 1/2$, no matter which pair of agents is chosen. For full credit, your algorithm should require 6 queries to the list of conversations. Justify the correctness of the algorithm as well as rigorously analyze the number of queries used by your algorithm.

Solution: Pre-Solution:

Based on the fact that the questions says "Armed with the knowledge from the previous parts", let's look through the previous parts to see which ones might be useful.

Let's start at the beginning, in the problem description. We know that each agent speaks with at least 3/4 of the agents from each of the other agencies. Does this mean that if the agents are from different agencies, there's at least a 3/4 chance that they spoke?

Unfortunately not, since the agents were not chosen randomly, but were instead given to us as the input. Our algorithm needs to work most of the time on every possible pair of agents, including agents from different agencies who haven't spoken to each other.

Next, let's look at part a. Does finding an agent a who is good for both of a_1 and a_2 help us tell if a_1 and a_2 are from the same agency? Not really, because it could be the case that they're from the same agency, and a is from a different agency, or it could be the case that they're from different agencies, and a is from the third agency.

What about b? If we find a special pair of agents $\{a, a'\}$, does that tell us anything about a_1 and a_2? It does, because as mentioned above, if an agent is good for a pair of agents from different agencies, that agent must be from the third agency, and two agents from the third agency can't have talked to each other.

So now we have a possible solution: Try and find a special pair of agents. There are 5 conversations involved in making a pair of agents a special pair, so we're only going to be able to check if one pair of agents is a special pair. Fortunately, we can apply part (c): If we pick two agents a and a' independently at random from the pool of remaining agents, if a_1 and a_2 are from the same agency, there is at least a $1/36$ chance that they are special. Note that we can apply probabilities here, where we couldn't before, because we are choosing a and a' randomly.

The algorithm so far looks like the following: Pick two random other agents, and query their conversation with each other and each of the conversations between them and a_1 and a_2. If all 5 of those conversations happened, then a_1 and a_2 are definitely from the same agency, and we can return SAME. Otherwise, we do something else.

We've done something pretty good, and we're low on remaining queries, so it'd be nice if the something else was simple. The simplest thing we could possibly do at this point would be to return a deterministic answer. Unfortunately, if we return SAME at this point, we'll never get it right in the case where a_1 and a_2 are from different agencies, while if we return DIFFERENT, we'll only get it right 1/36 of the time in the case where a_1 and a_2 are from the same agency. We could try to incorporate information about whether a_1

and a_2 have talked, but since they weren't chosen randomly, that doesn't help.

The next simplest thing we could do would be to return SAME with some probability p, and DIFFERENT with probability $(1-p)$. If we do this, our probability of success if the agents are from the same agency is at least $1/36$ (that we find a special pair), $+35/36 * p$ (in the case where we don't find a special pair).

Our probability of success if the agents are from different agencies is $(1-p)$.

A simple choice for p would be $1/2$. Then, in the same case, we would succeed with probability $1/36 + 35/72 = 37/72$, and in the different case, we would succeed with probability $1/2$. That's not quite good enough in the different case. So let's increase p a bit. If we choose $p = 0.49$, that should increase the different case a little, and decrease the same case a little, and give a chance of success of over $1/2$ in both cases, as desired.

Solution:

Given the input $\{a_1, a_2\}$, our algorithm works as follows:

(1) Choose a random pair of agents a and a' among the remaining agents.

(2) Check if a and a' are special (for $\{a_1, a_2\}$).

(3) If the check above succeeds, output "SAME".

(4) Otherwise, output "DIFFERENT' with probability $c = \frac{36}{71}$, "SAME' with probability $1 - c = \frac{35}{71}$.

Observe that that checking if a and a' are special for $\{a_1, a_2\}$ takes five list queries (we need to check pairs $\{a_1, a\}$, $\{a_1, a'\}$, $\{a_2, a\}$, $\{a_2, a'\}$, and $\{a, a'\}$). So, this algorithm indeed makes always five list queries (which is less than six). (We allowed a budget of six list queries as many people would probably feel urge to check if a_1 and a_2 spoken to each other, even though it is not needed to get the desired performance - do you see why?)

To analyze the correctness of this algorithm, notice first that if a and a' are special for $\{a_1, a_2\}$ then a_1 and a_2 have to belong to the same agency. This is so since, by Rule 1, if a and a' spoken to each other then they have to be in different agencies, but as *both* a and a' have to have spoken to *both* a_1 and a_2 too, it has to be the case that both a_1 and a_2 have to belong to different agency than as and a's ones. Clearly, as there is three agency in total, such situation can arise only if a_1 and a_2 are in the same agency.

Now, let us now lower bound the probability of our algorithm being correct. If a_1 and a_2 are in different agencies then, clearly, a and a' will never be special. This means we say "DIFFERENT" with probability $1 - c$.

On the other hand, if a_1 and a_2 are in the same agencies then, by part (c), with probability at least $\frac{1}{36}$ a and a' will be special, making us to correctly output "SAME". With the remaining, at most $1 - \frac{1}{36}$ probability, a and a' are not special and then we correctly output "SAME" with probability c. So, our total probability of being correct is at least

$$\frac{1}{36} + \left(1 - \frac{1}{36}\right)(1-c) = \frac{1}{36}(36 - 35c).$$

To figure out the right setting of c we used above, notice that we want in both cases the correct answer probability to be strictly larger than $\frac{1}{2}$. In other words, we want to maximize

$$\min\{c, \frac{1}{36}(36-35c)\}$$

and this minimum is maximized when these two quantities are equal. Consequently, the equation

$$c = \frac{1}{36}(36-35c)$$

implies, after rearranging and multiplying both sides by $36/71$, that

$$c = \frac{36}{71},$$

which makes both probabilities be greater than $\frac{1}{2}$, as desired.

(Of course, in your solutions, any value of c that makes these probabilities greater than $\frac{1}{2}$ would be sufficient.)

Problem 5. [23 points] **Triangle Counting** (3 parts)
In the triangle counting problem, we are given a graph $G = (V, E)$ with $n = |V|$ vertices, and we would like to estimate the number Δ_G of triangles in this graph. More precisely, we want to estimate

$$\Delta_G := |\{\{u, v, w\} \subseteq V \mid \text{all three edges } (u, v), (v, w) \text{ and } (u, w) \text{ belong to } E\}|.$$

Consider the following procedure for estimating the value of Δ_G.

1. Fix some T. (The value of T will be decided later.)
2. For $i \leftarrow 1$ to T:
   - Sample a triple $\{u, v, w\}$ of vertices *uniformly at random* from the set of all $\binom{n}{3}$ triples of vertices in G.
   - Set X_i be equal to $\binom{n}{3}$, if the vertices $\{u, v, w\}$ induce a triangle, i.e., if (u, v), (v, w) and (u, w) are all edges in G. Otherwise, X_i is equal to 0.
3. At the end, output an estimate
$$X := \frac{\sum_{i=1}^{T} X_i}{T}.$$

In what follows, we want to argue that, with high probability, X constitutes a good (approximate) estimate of Δ_G, provided T is sufficiently large.

(a) [7 points] Compute the expectation $\mathbb{E}[X]$ of the variable X.

Solution:
By linearity of expectation, we have

$$\begin{aligned} \mathbb{E}[X] &= \frac{1}{T}\sum_{i=1}^{T} \mathbb{E}[X_i] = \frac{1}{T}\sum_{i=1}^{T} \binom{n}{3} \Pr[\{u, v, w\} \text{ is a triangle}] \\ &= \frac{\binom{n}{3}}{T}\sum_{i=1}^{T} \frac{\Delta_G}{\binom{n}{3}} = \Delta_G, \end{aligned}$$

where we used the fact that $\Pr[\{u, v, w\} \text{ is a triangle}] = \frac{\Delta_G}{\binom{n}{3}}$ as we choose the set $\{u, v, w\}$ uniformly at random among all the $\binom{n}{3}$ possible vertex triples.

(b) [7 points] What is the variance $\mathrm{Var}(X)$ of the variable X?
Hint: Recall that Var $(Y) = \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2$.

Solution:
Let us first notice that each X_i is identically distributed and independent. So, for any $1 \le i \le T$, we have that

$$\mathbb{E}\left[X_i^2\right] = \binom{n}{3}^2 \Pr\left[\{u,v,w\} \text{ is a triangle}\right] = \binom{n}{3} \cdot \Delta_G,$$

where again we used the fact that $\Pr\left[\{u,v,w\} \text{ is a triangle}\right] = \frac{\Delta_G}{\binom{n}{3}}$. (Note that this probability indeed does not depend on i in any way.)
On the other hand, by analogous calculations, for each $1 \le i \le T$, we have that

$$\left(\mathbb{E}\left[X_i\right]\right)^2 = \left(\binom{n}{3} \Pr\left[\{u,v,w\} \text{ is a triangle}\right]\right)^2 = \Delta_G^2.$$

Now, we can conclude that, by the fact that all X_is are independent, we have that

$$\begin{aligned} Var\left(X\right) &= \frac{1}{T^2}\sum_{i=1}^{T} Var\left(X_i\right) = \frac{1}{T^2}\sum_{i=1}^{T}\left(\mathbb{E}\left[X_i^2\right] - \left(\mathbb{E}\left[X_i\right]\right)^2\right) \\ &= \frac{1}{T^2}\sum_{i=1}^{T}\left(\binom{n}{3} \cdot \Delta_G - \Delta_G^2\right) \\ &= \frac{\Delta_G}{T} \cdot \left(\binom{n}{3} - \Delta_G\right). \end{aligned}$$

(c) [9 points] What should be the value of the parameter T in order to ensure that, with probability at least 0.99, the value of X is within 1% of the number Δ_G of triangles in G?

Hint: The Chebyshev inequality might be helpful here. Recall that this inequality states that for any random variable Y and any $k > 0$,

$$\Pr\left[\left|Y - \mathbb{E}\left[Y\right]\right| > k\sqrt{\mathrm{Var}(Y)}\right] < \frac{1}{k^2}.$$

Solution:
Applying Chebyshev's inequality with $k = 10$, we obtain that

$$\Pr\left[\left|X - \mathbb{E}\left[X\right]\right| > 10 \cdot \sqrt{Var(X)}\right] = \Pr\left[\left|X - \Delta_G\right| > 10 \cdot \sqrt{\frac{\Delta_G}{T} \cdot \left(\binom{|V|}{3} - \Delta_G\right)}\right] < \frac{1}{100},$$

where we used the calculations from (a) and (b).

Now, we need to choose a value of t such that

$$10 \cdot \sqrt{\frac{\Delta_G}{T} \cdot \left(\binom{|V|}{3} - \Delta_G\right)} \leq \frac{1}{100}\Delta_G,$$

for all values of Δ_G. (Note that $0 \leq \Delta_G \leq \binom{|V|}{3}$.)
The above condition always holds when $\Delta_G = 0$ and otherwise is equivalent to

$$10^6 \cdot \frac{\left(\binom{|V|}{3} - \Delta_G\right)}{\Delta_G} \leq T.$$

The expression on the left is maximized for $\Delta_G = 1$ making the optimal choice of T be

$$T = 10^6 \cdot \left(\binom{|V|}{3} - 1\right).$$

Note: In principle, we don't want the value of T to depend on the (unknown) value of Δ_G, but it is ok if in your solution T does depend on it.

Problem 6. [16 points] **Expected Utility of Nash Equilibrium** (2 parts)

(a) [6 points] Give an example of a two-person game such that the following two conditions are both satisfied:

1. It has at least two *different* Nash equilibria $(\hat{x}, \hat{y})$ and $(\tilde{x}, \tilde{y})$;
2. The payoff of the first player is *different* when the Nash equilibrium $(\hat{x}, \hat{y})$ is played and when the Nash equilibrium $(\tilde{x}, \tilde{y})$ is played.

Hint: Your construction does not need to be complicated.

Solution:
Pre-Solution:

In general, mixed Nash Equilibria are tougher to deal with, so we want to specify a game that has pure Nash Equilibria. Also, if we can do this in the simplest case where there are 2 possible strategies per player, that would be better.

In a Nash Equilibrium, neither player has an incentive to change strategies. Thus, we want it to be the case that if the two players choose a specific pair of pure strategies, deviating from that pair of pure strategies is disadvantageous for both players.

Solution:

Consider a game

$$\begin{bmatrix} (1,1) & (0,0) \\ (0,0) & (2,2) \end{bmatrix}$$

Observe that this graph has two Nash equilibria: one corresponding two both players playing their first action, and one corresponding to both players playing their second action. (None of the players wants to unilaterally deviate as, in each case, it would reduce their payoff to 0.) The payoff of the first player is 1 and 2 respectively, in these Nash equilibria, as desired.

(b) [10 points] Prove that all the Nash equilibria of *any* two-player *zero-sum* game result in the same expected utility for the row player. In other words, show that for any two Nash equilibria $(\hat{x}, \hat{y})$ and $(\tilde{x}, \tilde{y})$ of this type of game described by a (row player) payoff matrix A, we have

$$\hat{x}^T A\hat{y} = \tilde{x}^T A\tilde{y}.$$

Hint: Consider any Nash equilibrium $(\hat{x}, \hat{y})$.
Relate the value of $\hat{x}^T A\hat{y}$ to $V_R := \max_x \min_y x^T Ay$ and $V_C := \min_y \max_x x^T Ay$.
(Remember the intuition behind these values.) Then, use the MinMax theorem appropriately.

Solution:
Pre-Solution:

We start with what we know - the definition of a Nash Equilibrium. In particular, player 1 does not have an incentive to deviate from his strategy in a Nash Equilibrium. In other words, given the knowledge that player 2 will choose the strategy $\hat{y}$, the expected payoff of player 1 can not get any better if he uses a strategy other than the strategy $\hat{x}$ corresponding to the Nash Equilibrium.

We want to eventually relate the expected payoffs of Nash Equilibria to the quantities $V_R := \max_x \min_y x^T Ay$ and $V_C := \min_y \max_x x^T Ay$, so we try to write down expressions about these Nash Equilibria that look related to V_R and V_C. Then, we will see if we can use math to relate the expected payoffs to V_R and V_C. Finally, we will try to somehow take advantage of the MinMax theorem that tells us that $V_R = V_C$.

Solution:

Consider a Nash equilibrium $(\hat{x}, \hat{y})$ of this zero-sum game. Note that the fact that $(\hat{x}, \hat{y})$ is a Nash equilibrium implies that

$$\hat{x}^T A\hat{y} = \max_x x^T A\hat{y}, \tag{1}$$

and that

$$\hat{x}^T A\hat{y} = \min_y \hat{x}^T Ay, \tag{2}$$

as otherwise either the row player or the column player would have an incentive to deviate. Now, (1) implies that

$$\hat{x}^T A\hat{y} = \max_x x^T A\hat{y} \geq \max_x \min_y x^T Ay = V_R, \tag{3}$$

where we use the simple fact that for any $\hat{y}$ and any x, $x^T A\hat{y} \geq \min_y x^T Ay$. Similarly, (2) gives us that

$$\hat{x}^T A\hat{y} = \min_y \hat{x}^T Ay \leq \min_y \max_x x^T Ay = V_C. \tag{4}$$

So, putting (3) and (4) together, we get that

$$V_R \leq \hat{x}^T A\hat{y} \leq V_C.$$

But, by the MinMax theorem we know that $V_R = V_C = V$ and thus it must be that $\hat{x}^T A\hat{y} = V$. So, as V does not depend on $(\hat{x}, \hat{y})$ in any specific way, it must be the case that indeed $\hat{x}^T A\hat{y} = \tilde{x}^T A\tilde{y}$, as long as both $(\hat{x}, \hat{y})$ and $(\tilde{x}, \tilde{y})$ are Nash equilibria.