CS3230 Midterm Solutions

1. (a) False

   Idea: $\lim_{n\to\infty} \frac{n!}{2022^n} = \lim_{n\to\infty} \frac{1}{2022} \cdot \frac{2}{2022} \cdot \dots \cdot \frac{n}{2022} = \infty$, so $n! \neq O(2022^n)$.

   (b) False

   Idea: $4^{\lg n} = (2^2)^{\lg n} = (2^{\lg n})^2 = n^2$, so $n^2 \neq o(4^{\lg n})$.

   (c) True

   Idea: For constants $c > d > 0$, we have $n^c = \omega(n^d)$.

   (d) True

   Idea: $1 + \frac{1}{2} + \dots + \frac{1}{n} \approx \ln n = \Theta(\lg n)$, which is also $\Omega(\lg n)$.

   (e) False

   Idea: $\lim_{n\to\infty} \frac{n^{n-1}}{n^n} = \lim_{n\to\infty} \frac{1}{n} = 0$, so $n^{n-1} \neq \Theta(n^n)$.

   (f) False

   Idea: As discussed in Lecture 2, the running time of this recursive program is exponential in n.

   Rubric: 2 points for each correct answer, 1 point for each "Don't know".

   Comment: There were some mistakes, but the overall performance on this question was quite good.

2. (a) $T(n) = \Theta(n^2)$

   Idea: $T(n) = 3230(n + (n-1) + \dots + 1) = 3230 \cdot \frac{n(n+1)}{2} = \Theta(n^2)$.

   Comment: Some answers $\Theta(n)$ or $\Theta(n^3)$, but mostly correct.

   (b) $T(n) = \Theta(n^3)$

   Idea: Case 3 of master method with $a = 7$, $b = 2$, $f(n) = n^3$.

   Comment: Some answers $\Theta(n^{\lg 7})$, but mostly correct.

   (c) $T(n) = \Theta(\lg^2 n)$

   Idea: Case 2 of master method with $a = 1$, $b = 3/2$, $f(n) = \lg n + 1 = \Theta(\lg n)$

   Comment: Many answers $\Theta(\lg n)$ or $\Theta(\lg\lg n)$.

   (d) $T(n) = \Theta(\sqrt{n})$

   Idea: Using the recursion tree, we need to do $n^{1/2}$ work in the first level, $n^{1/4}$ work in the second level, and so on. This decreases rapidly, so we might guess that $T(n) = \Theta(\sqrt{n})$. We verify our guess using the substitution method, by claiming that $T(n) \leq c\sqrt{n}$ for some constant c. If this holds for $T(\sqrt{n})$, then by our recurrence, $T(n) \leq cn^{1/4} + \sqrt{n}$; in order for the right-hand side to be at most $c\sqrt{n}$, we need $n^{1/4} \geq \frac{c}{c-1}$. Hence, we may choose c large enough so that this inequality holds for all $n \geq 2$ and also $c \geq T(1)$ for the base case. So $T(n) = \Theta(\sqrt{n})$.

   Comment: Few correct answers. $\Theta(\sqrt{n}\lg\lg n)$ was a common mistake using recursion tree, and $\Theta(n)$ was a common incorrect guess.

   Rubric for (a)–(d): 2 points each for (a) and (b), 3 points each for (c) and (d). No partial credit. For part (c), writing as $\Theta((\lg n)^2)$ is fine; $\Theta(\lg\lg n)$ is not.

   (e) $\dfrac{1}{6e}$

   Idea: Proceed similarly to Assignment 5 Question 2(a). The probability that a particular bin contains exactly three balls is $\binom{n}{3}\left(\frac{1}{n}\right)^3\left(\frac{n-1}{n}\right)^{n-3} = \frac{n(n-2)}{6(n-1)^2} \cdot \left(1 - \frac{1}{n}\right)^n$. Since $\lim_{n\to\infty} \frac{n(n-2)}{6(n-1)^2} = \frac{1}{6}$ and $\lim_{n\to\infty}\left(1 - \frac{1}{n}\right)^n = \frac{1}{e}$, the desired limit is $\frac{1}{6e}$.

   Rubric: 4 points for the correct answer. Partial credit in the following cases:

   - 1 point for an answer of the form $\frac{c}{e}$ for some constant c (e.g., $\frac{1}{e}$, $\frac{3}{e}$, or $\frac{1}{3e}$)

- 2 points for finding the correct expression $\binom{n}{3}\left(\frac{1}{n}\right)^3\left(\frac{n-1}{n}\right)^{n-3}$
  - 1 point if the expression is incorrect but contains the term $\left(\frac{1}{n}\right)^3\left(\frac{n-1}{n}\right)^{n-3}$

The partial credits can be combined.

Comment: Only a few arrived at $\frac{1}{6e}$. Some answered $\frac{1}{6}$ or $\frac{1}{e}$. Many dropped the term $\binom{n}{3}$, resulting in a limit of 0.

(f) $\dfrac{2023}{4044}$

Idea: In order for $X = x_1$ at the end, X must be unchanged throughout the whole process. The probability that this happens is $\left(1-\frac{1}{2^2}\right)\left(1-\frac{1}{3^2}\right)\dots\left(1-\frac{1}{2022^2}\right) = \frac{1\cdot 3}{2^2}\cdot\frac{2\cdot 4}{3^2}\cdot\dots\cdot\frac{2021\cdot 2023}{2022^2} = \frac{2023}{4044}$.

Rubric: 4 points for the correct answer (no point deducted if the fraction is not simplified as long as the numbers are not too large, e.g., $\frac{4046}{8088}$). Partial credit in the following cases:

- 1 point if the answer (after simplifying the fraction) has numerator 2022, 2023, or 2024, **and** denominator 4042, 4044, or 4046
- 2 points for finding the correct product expression $\left(1-\frac{1}{2^2}\right)\left(1-\frac{1}{3^2}\right)\dots\left(1-\frac{1}{2022^2}\right)$, which may also be written as $\prod_{i=2}^{2022}\left(1-\frac{1}{i^2}\right)$
  - 1 point if both indices are off by at most one (e.g., starts at $i = 3$ instead of $i = 2$ and/or ends at $i = 2021$ instead of $i = 2022$)

The partial credits can be combined.

Comment: Half the students had the correct product form, but not many carried out the calculation. Some drew the incorrect analogy from Assignment 6 Question 3 and answered $\frac{1}{2022^2}$.

3. As in the hint, first consider the family $\mathcal{H}_{\text{all}}$ of all functions from N to M. We claim that for any $x, y \in N$,

$$\Pr_{h\in\mathcal{H}_{\text{all}}}[h(x) = h(y)] = \frac{1}{m}.$$

Indeed, for any $(i, j) \in M \times M$, the number of functions such that $h(x) = i$ and $h(y) = j$ is exactly the same as for any other (i', j'). Since there are m^2 pairs (i, j), and $h(x) = h(y)$ for exactly m of them, the probability that $h(x) = h(y)$ is $m/m^2 = 1/m$. More specifically, there are m^n functions in $\mathcal{H}_{\text{all}}$, and $h(x) = h(y)$ for m^{n-1} of them.

Next, modify $\mathcal{H}_{\text{all}}$ slightly by removing the function that maps all $w \in N$ to 1. Call the new family $\mathcal{H}_{\text{almost}}$. There are $m^n - 1$ functions in $\mathcal{H}_{\text{almost}}$, and for any $x, y \in N$, $h(x) = h(y)$ for $m^{n-1} - 1$ of these functions. Hence,

$$\Pr_{h\in\mathcal{H}_{\text{almost}}}[h(x) = h(y)] = \frac{m^{n-1}-1}{m^n-1} < \frac{m^{n-1}}{m^n} = \frac{1}{m},$$

as desired.

Note: It is not necessary to derive the numbers m^n and m^{n-1}. One could also assume that there are b functions in $\mathcal{H}_{\text{all}}$, and $h(x) = h(y)$ for a of them, where $a/b = 1/m$. Then, in $\mathcal{H}_{\text{almost}}$, there are $b - 1$ functions, and $h(x) = h(y)$ for $a - 1$ of them, so the probability that $h(x) = h(y)$ is $\frac{a-1}{b-1} < \frac{a}{b} = \frac{1}{m}$.

Note: Another possibility is to remove all m constant functions. This leads to the calculation $\frac{m^{n-1}-m}{m^n-m} < \frac{m^{n-1}}{m^n} = \frac{1}{m}$.

Rubric: 10 points in total, broken down as follows:

- 3 points for claiming that $\Pr_{h\in\mathcal{H}_{\text{all}}}[h(x) = h(y)] = \frac{1}{m}$
- 2 points for justifying the above bullet point
  - 1 point can be given if a justification is provided but not completely satisfactory

- 3 points for suggesting a valid modification of $\mathcal{H}_{\text{all}}$
- 2 points for justifying the above bullet point
  - 1 point can be given if a justification is provided but not completely satisfactory

The partial credits below can be combined with each other, but not with the bullet points above.

- 2 points for proving the statement for some $m \geq n \geq 2$ (a singleton family suffices in this case)
- 3 points for proving the statement for some $n > m \geq 2$ (e.g., for $n = 3$ and $m = 2$)
- 1 point for mentioning universality
- 1 point for mentioning that $\mathcal{H}_{\text{all}}$ contains m^n functions
- 1 point for mentioning universality

Comment: Despite the hint, only a few students solved this question—apparently the hint was too difficult to understand or use properly. Many tried to use linearity of expectation or pairwise independence, which are unhelpful. A couple of students used the matrix construction from Lecture 5 and removed the hash function corresponding to the all-0 matrix—this works, but since the matrix construction is only for n, m that are powers of two, 8 points were awarded.

4. (a) Like in the second-largest element algorithm, arrange the players in a balanced knockout tournament structure, and make queries according to the matches that arise in this tournament. The total number of matches is $15 - 1 = 14$, and the winner of the tournament (say, P_1) has played against at least three other players (say, P_2, P_3, P_4). Note that all players besides P_1 have lost at least one match each, so P_1 is the only remaining candidate. Query the outcomes between P_1 and $P_5, P_6, \ldots, P_{15}$. If P_1 beats all of these players, then P_1 beats all other 14 players, so we answer "Yes". Else, no player beats all other players, and we answer "No". The total number of questions is $14 + 11 = 25$.

   Rubric: 10 points in total, given according to the number of questions needed:

   - 10 points for 25 questions
   - 8 points for 26 questions
   - 7 points for 27 questions (e.g., doing a simple pass using 14 questions, noting that the winner must have played against at least one other player, and query the matches between this winner and the 13 remaining players)
   - 5 points for 28 questions (e.g., doing a simple pass using 14 questions and query the matches between the winner and the 14 remaining players)
   - 4 points for 29–30 questions
   - 3 points for 31–50 questions
   - 2 points for 51–70 questions
   - 1 point for 71–104 questions
   - 0 points for 105 questions (i.e., asking all match outcomes)
   - The idea for the first stage of the algorithm alone (with 14 questions) is worth 2 points.

   Points can be deducted for the following (the deductions can be combined):

   - No or unsatisfactory justification of correctness. If the correctness is deemed to be "easy" (such as in the 25-, 27-, or 28-question algorithms above), the deduction will be no more than 2 points. Otherwise, the deduction can be adjusted appropriately.
   - Incorrect count on the number of questions.
     - If the error is a simple arithmetic error (e.g., incorrectly adding up the number of questions), the deduction will be 1 point.

  - If the error stems from more advanced reasoning (e.g., claiming that P_1 has played against at least *four* other players in the knockout tournament and thus ending up with $14+10=24$ questions), the deduction will be 2 points.

Comment: A wide range of progress was made on this question and graded according to the rubric. Some students used the maximum algorithm from Lecture 1 but did not realize that the winner from this algorithm may not win against all players due to the lack of transitivity. A fairly common mistake was to reason that P_1 has played against at least four other players in the knockout tournament, resulting in 24 questions—2 points were deducted for this.

(b) We claim that you need to ask for all $\binom{15}{2} = 105$ match outcomes. To show this, we use an adversary argument. Consider the outcome where each P_i beats $P_{i+1}, P_{i+2}, \ldots, P_{i+7}$ and loses to $P_{i+8}, P_{i+9}, \ldots, P_{i+14}$. (If $k > 15$, then P_k means P_{k-15}.) Note that this outcome is *balanced*: every player beats 7 other players and loses to the remaining 7 players. The adversary answers questions according to this balanced outcome.

Suppose that you have not queried all matches. It is possible that the outcome is balanced, in which case all 15 players win the highest number of matches. But if the outcome of any unqueried match is reversed, then only one player wins the highest number of matches (i.e., 8 matches). Hence, no matter what you answer, the adversary can make sure that this answer is wrong.

Rubric: 10 points in total, broken down as follows:

- 1 point for claiming that all matches need to be queried (i.e., the answer is 105)
- 1 point for mentioning "adversary argument" or "adversary"
- 4 points for finding a balanced outcome, where all players win the same number of matches
- 4 points for completing the argument assuming the existence of a balanced outcome
  - 2 points if the adversary's strategy is specified, but no proof of correctness is given

If the argument works only against a non-adaptive algorithm, 5 points are awarded. Partial credit can be awarded if an argument is attempted but not correct.

Comment: A fair number of students made the correct guess of 105 and/or realized that an adversary argument should be used; however, not many solved the problem. Some students used an adversary argument that works only against a non-adaptive algorithm (e.g., "Assume the algorithm queries all matches except the match between P_1 and P_2. The adversary makes P_1, P_2 win against all remaining players, so the winner of the P_1 vs P_2 match is the only player with the highest number of wins"). For this problem, it is crucial that the algorithm can be adaptive—otherwise in part (a) we would need all 105 questions instead of only 25. Hence, 5 points were awarded for such a solution, and 3–4 points were given for similar but incomplete attempts. Some students tried to use the decision tree, which is not the right tool for this problem and can only yield a weak bound.