ADA 2022
Midterm Review

17th Nov, 2022

Agenda

- Statistics
- Regrade Request
- Highlights

Statistics

Main Statistics

(before regrade requests)

- Mean: 52.26
- Median: 50.50
- Standard Deviation: 15.82
- For other statistical analysis, please refer to the file attached in the announcement.

Congratulations to Top 10

(before regrade requests)

黃仲群	108.0	李沛宸	91.0
王褕立	101.5	侯欣緯	90.5
林伯禧	99.5	陳楷元	90.5
吳柏燁	96.0	林煜傑	89.0
賴昭勳	95.5	吳柏翰	85.5

Regrade Request

Steps & Schedule

- Regrade request is now available.
- The UI for Gradescope is straightforward.
- Duration: 2 weeks
  - 17th Nov 2022 - 30th Nov 2022
- FCFS: send your regrade requests ASAP

Tips

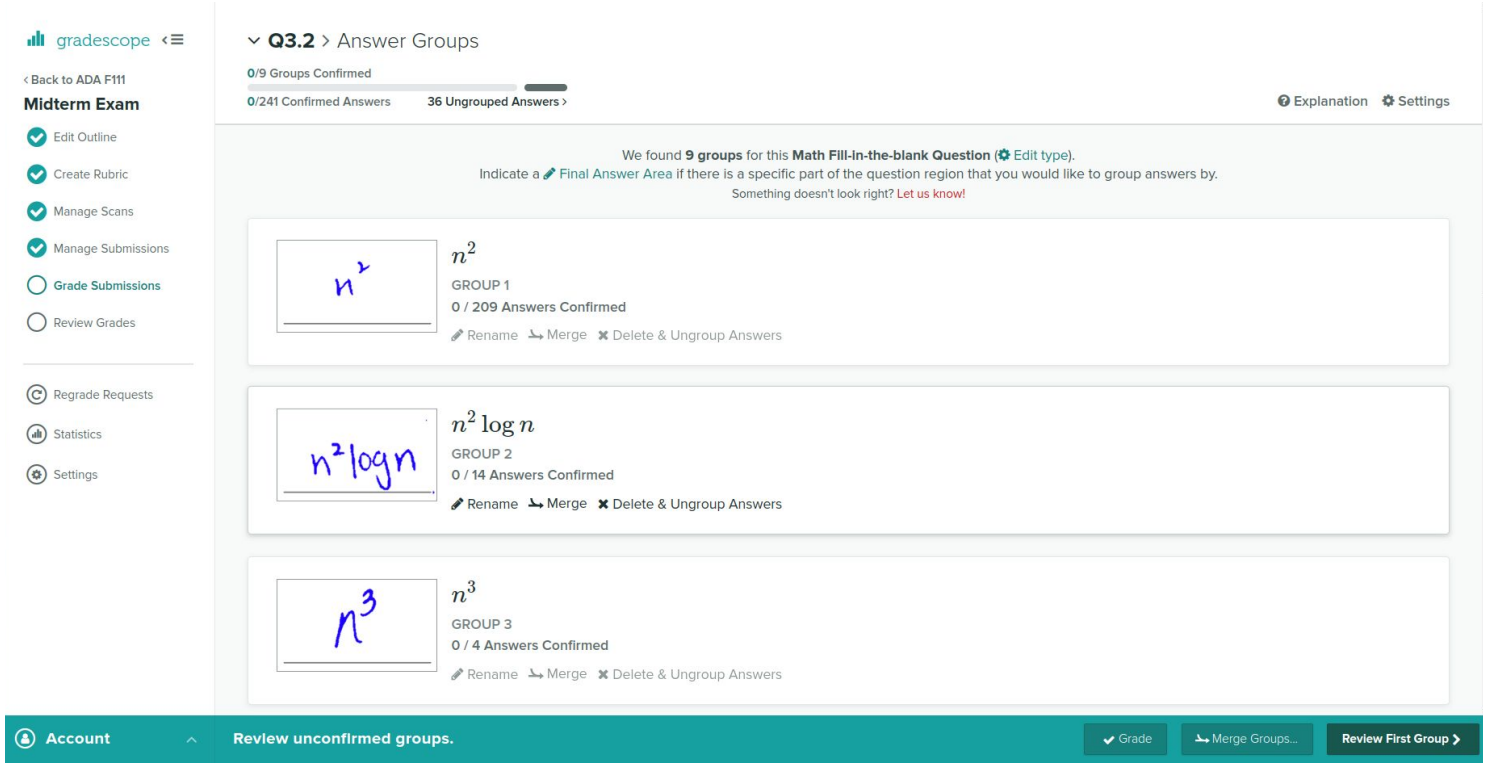
- If you were the TA grading this problem, ...?
- Write down your arguments in bulleted / ordered lists.
- Explain your reasoning concretely.
    - DONTs: “I think I should get more points for this.”
    - DOs: “I think I can get back the 1 point for lacking ... since writing ... could be equivalent to ...”

Highlights

AI-Assisted Grading

- Applied on Problem 1, Problem 3-(a)-(ii) and Problem 6-(a)
- Send regrade requests if your handwriting is incorrectly identified.
- Please use a pen with darker color if you receive an email later.

AI-Assisted Grading



AI-Assisted Grading

Problem 3(b).

MINIMUM	MEDIAN	MAXIMUM	MEAN	STD DEV
0.0	4.0	5.0	3.08	2.1

RUBRIC	POINTS	PERCENTAGE OF STUDENTS
Correct	+ 0.0	20%
Did not specify induction step (suppose for all n<k it holds...)/Wrong induction step (n=k to n=k+1 ...)	+ 0.0	36%
No induction base case	- 1.0	17%
Other minor mistake	- 0.5	9%
Other major mistake/many minor mistake	- 1.0	8%
Only choose the correct c that is proveable./Correct base case	- 4.0	9%
Wrong/Did not finish your proof	- 5.0	15%

Common Mistake

- You cannot change the c you have chosen during your proof
- For example, we can prove $2^n = O(n)$ by

Prove : $2^n = O(n)$, which means $2^n \leq cn, \forall n > 1, c > 0$
Base Case : $2^1 = O(1)$
Inductive Step :
Suppose for $n = k - 1$ we have $2^{k-1} \leq c(k-1)$
When $n = k, 2^k = 2 \cdot 2^{k-1} \leq 2c(k-1)$, take $c' = 2c$ we have $2^k \leq c'(k-1) \leq c'k$
Which is wrong.

Problem 4 - Why Is It?

- The grader cannot come today.
- Let’s read his words.

Problem 5.

- A dynamic programming algorithm should contain:
  - State
  - Transition function

Problem 5 - State.

- Please give a clear statement to your state.
  - Bad: dp[i] := end at i
    - What do you store in your dp table?
    - What is “end”?
  - Good: dp[i] := Consider only whacking the **1~i-th** moles, **the maximum score** BB can get.
    - Enough clear.
    - However, this state definition doesn’t have an optimal substructure.

Problem 5 - State.

- dp[i] := Consider only whacking the **1~i-th** moles, **the maximum score** BB can get.
  - Why no optimal substructure?
- No matter how we enumerate the case, we still cannot know where the “hand” is. We cannot verify the achievability of whacking.
- Correct: Consider only whacking the 1~i-th moles, and BB’s has **whacked the i-th mole**, the maximum score BB can get.

Problem 5 - Transition Function.

- Common mistake:
  - Include dp[i - 1] in dp[i].
    - Same problem before.
  - Whack the same mole twice.
  - Greedly choose a mole to whack.
- Redundant base cases (no point deduction)
  - Base cases are terminate statements. Just making sure your recursion always terminates is enough.

Problem 5 (c).

- This is not a dynamic programming problem.
- Instead, it provides a clue, leading to the time complexity optimization in problem 5 (d).
    - Problem 5 (d) is considered a difficult problem.

Problem 5 - Grader’s Words.

- BB is not the problem setter instead of the grader.
- BB doesn’t show off in front of his girlfriend in Tom’s world.
- Some of you give unclear statements and hence BB cannot understand your words.
    - If you believe you have met the specification, please send regrade requests to explain it.

Problem 6.

- I’m curious about how many of you actually read this problem
- Don’t be afraid of long descriptions* and cute symbols!
- *We’ve tried our best to make it simple
- Contact me (or TAs) if you want to see the full answer

Problem 6(a)

- 4 multiplications
- 6 additions

```
   67          63
x  89       +  54
-----       ------
   63          603
  54       +  56
  56       ------
 48           1163
-----       + 48
 5963       ------
             5963
```

Problem 6(b)

- We demonstrate subtask 1 here
- carrying

Notice that $[a, b] \times [c, d] = [ac, ad + bc, bd]$ (do carrying after this in $O(N)$), However, after computing ac and bd, we can obtain $ad + bc$ by

$$(a + b)(c + d) - ac - bd$$

Problem 6(c)

Hint: Recall that you can use previous results no matter if you have solved it.

(i) It runs in $O(N^{\log_2 3})$ time.

(ii) For any $\epsilon > 0$, show that there exists an algorithm that runs in $O(N^{1+\epsilon})$.

$$f(N) = (2k-1)f\left(\frac{N}{k}\right) + O(N)$$

$$f(N) = O(N^{\log_k(2k-1)})$$

Problem 6(d)

- This is actually a good problem

Give a proof to show that

$$N2^{10\sqrt{\log N}} = o(N^{1.01})$$

Lemma 1 *If* $f(N) = o(g(N))$ *and* $\lim_{N\to\infty} g(N) = \infty$, *then*

$$2^{f(N)} = o(2^{g(N)})$$

Problem 6(d) - cont.

- 1.5 points for $\log(LHS) = o(\log(RHS))$
- 1.5 points for proving $f = o(g) \Rightarrow 2^f = o(2^g)$ if $g \to \infty$.
  - In other words, if someone just claimed he's done with $\log(LHS) = o(\log(RHS))$, then he will get 1.5 points only.
  - I think this step is not trivial, since:
    - $f = O(g) \Rightarrow 2^f = O(2^g)$ is wrong (ex: $f = 2x, g = x$)
    - $f = o(g) \Rightarrow 2^f = o(2^g)$ is wrong (ex: $f = x^{-1}, g = 1$)
- If someone proves big-O only, he will get 1 points

Problem 6(e)

$O(N2^{c\sqrt{\log N}})$

- Give a guess!

$$f(N) = (2k-1)f\left(\frac{N}{k}\right) + O\left(k^2 \cdot \frac{N}{k}\log k\right)$$

- The answer is

$$k = 2^{d\sqrt{\log N}}$$

Problem 6 - Fun facts

- The average score for 6(e) is 0.00159, with the highest (and only) score 0.5 by 吳柏燁
- halloween math

Problem 7 - How Is It?

- Thanks so much for the constructive feedback.
- 5-6 Things: 3 points
- 3-4 Things: 2 points
- 1-2 Things: 1 point
- Nothing: 0 point

Problem 7 - How Is It?

Problem 7 - How is it?

(a) 3 things you like about this course

(i)

自從修了ADA之後，我都不用去
健身房就可以維持強壯的身心了

(ii)

自從修了ADA後，我生活都不會無聊，
每天就是起床寫作業直到晚上睡覺

(iii)

我變得很會管理時間，把DP用在管理
寫各科作業的時間，真棒

Q & A
Thank you!