CS498 – M14/MZ4 — "MCS Algorithms"

http://courses.grainger.illinois.edu/cs498m14/fa2026

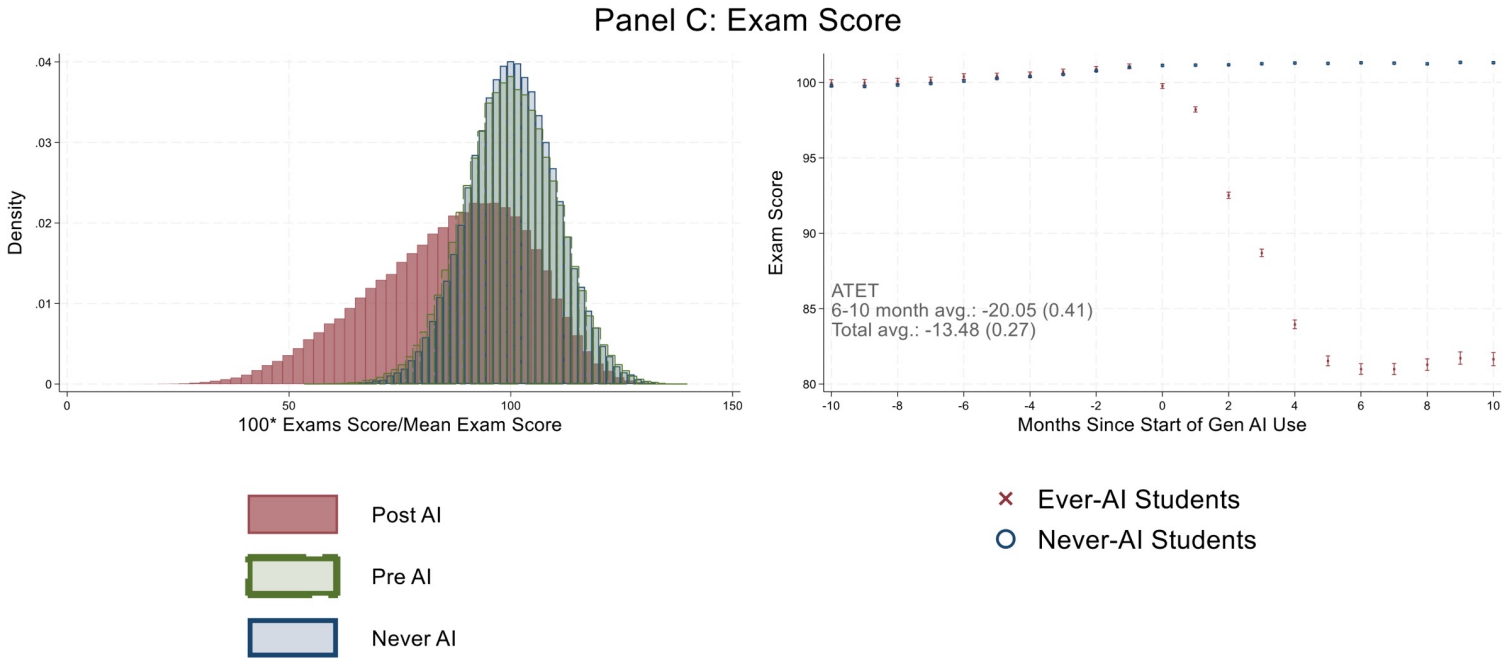
Specifically aimed at MCS students
without prior theoretical algorithms class like 374

Two background assessment

But AI tho!

Weekly HW	30%	Tue 9pm	Wed 9pm 80%
MT1	20%		
MT2	20%		
Final exam	30%		

Strömberg, Lei, Wu (2026)



Yong, Nacenta, Estay (2026)

6.2 Self-efficacy and Metacognition with GenAI

When looking at the results about RQ1 (self-efficacy) and RQ2 (learning) in conjunction, we find the evidence alarming: GenAI increases student's confidence about their learning more than LT, yet it produces the worst learning results out of the three learning supports, as measured by our quiz.

This could be interpreted as a failure of metacognition, which has been identified as one of the biggest challenges introduced by GenAI in general [72], and for learners in particular [70]. Our analysis indicates that this challenge extends to more advanced CS classes, despite our suspicion that more advanced learners of algorithms would be more mindful about their conceptual learning, as opposed to in introductory programming where it might be easier to forgive students for thinking that the more low-level and mechanical aspects of the programming tasks are less demanding or less worthy of effort. Nevertheless, not everything is a failure of metacognition since, on average, participants seemed to prefer the LT condition for most aspects of the task when they had experienced them all.

Algorithms

Explicit, precise, unambiguous, mechanically executable instructions — usually for some task

Sing "n bottles of beer on the wall"

BOTTLESOFBEER(n):

For $i \leftarrow n$ down to 1

Sing "*i bottles of beer on the wall, i bottles of beer,*"

Sing "*Take one down, pass it around, $i-1$ bottles of beer on the wall.*"

Sing "*No bottles of beer on the wall, no bottles of beer,*"

Sing "*Go to the store, buy some more, n bottles of beer on the wall.*"

$\Theta(n)$ time # lines or # words

singing x takes $\Theta(\log x)$ syllables

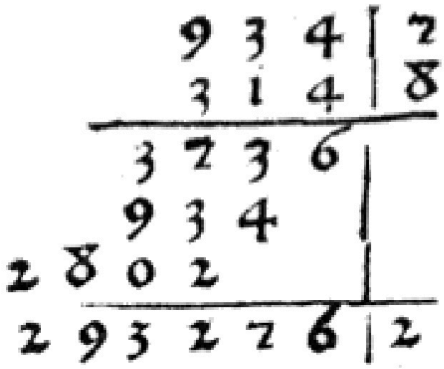
$\Rightarrow \Theta(n \log n)$ syllables

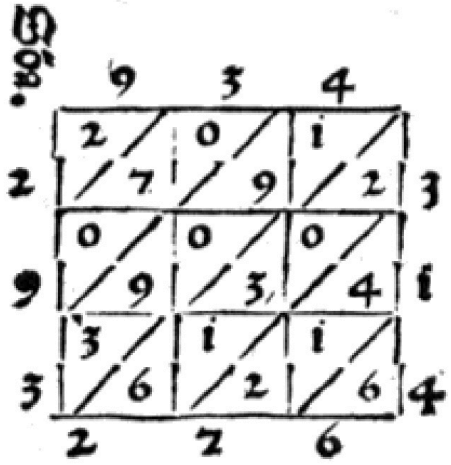
Algorithm ← al-Khwarizmi

algorism ← algoritmi

αριθμος = number

αλγος = pain





```
FIBONACCIMULTIPLY(X[0..m-1], Y[0..n-1]):
  hold ← 0
  for k ← 0 to n + m - 1
      for all i and j such that i + j = k
          hold ← hold + X[i] · Y[j]
      Z[k] ← hold mod 10
      hold ← ⌊hold/10⌋
  return Z[0..m + n - 1]
```

O(mn) time

iterative algorithm

How do I know this is correct?

$$x = \sum_{i=1}^{\hat{}} X[i] \cdot 10^i \qquad y = \sum_{j=1}^{m} Y[j] \cdot 10^j$$

$$\Rightarrow \quad x \cdot y = \sum_{i=1}^{n} \sum_{j=1}^{m} X[i] \cdot Y[j] \cdot 10^{i+j}$$

+ induction

Russian peasant / duplation+mediation / Egyptian

```
PEASANTMULTIPLY(x, y):
  prod ← 0
  while x > 0
      if x is odd
          prod ← prod + y
      x ← ⌊x/2⌋
      y ← y + y
  return prod
```

x	y	prod
123	456	0 → 456
61	912	→ 1368
30	1824	
15		
7		
3		
1		
0		

#digits in y ≤ #digits of y initially + #iters in middle

$$x \cdot y = \begin{cases} 0 & \text{if } x = 0 \\ \lfloor x/2 \rfloor \cdot (y+y) & \text{if } x \text{ is even} \\ \lfloor x/2 \rfloor \cdot (y+y) + y & \text{if } x \text{ is odd} \end{cases}$$

induction

$$\#\text{iterations}(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 + \#\text{iters}(\lfloor x/2 \rfloor) & \text{o/w} \end{cases}$$

$= O(\log x)$

$T(n) = O(1) + T(\frac{n}{2})$

Each iteration takes $O(\log x + \log y)$ time

$\Rightarrow$ Total time $= O(\log x \cdot \log y)$ if $x \le y$

$= O(nm)$ if x has n digits, y has m digits