CSCI 0500: Data Structures, Algorithms, and Intractability (Fall 2026)

Assignment 0

Due at 11:59pm ET, Friday, Sep 18

1. (1 point) Review the first lecture and the syllabus PDF and answer the following questions.

   (a) Alice is unable to submit an assignment or workshop on time due to network issues. She emails her solutions to the professor two minutes after the deadline, and the files are timestamped before the deadline. What score will Alice receive?

   (b) Alice and Bob work together on an assignment or workshop problem on a whiteboard. They take a photo of the whiteboard, and then each writes up their own solution using the photo as a reference and submits it. Is this a violation of course policy?

   (c) Bob asks Alice to share her solution to an assignment, promising to use it only for learning purposes. Alice agrees. Bob understands Alice's solution, writes and submits his solution, and can reproduce it independently. Is this a violation of course policy?

   (d) Alice uses an AI tool to obtain a solution to an assignment problem. She understands the solution and can reproduce it independently. She directly copies the AI-generated response into her submission. Is this a violation of course policy?

   (e) Suppose the course has 8 workshops (each out of 1 point), 8 assignments (each out of 3 points), and 8 labs (each out of 1 point). Alice scores $(1, 1, 1, 1, 0, 0, 1, 1)$ on the workshops, $(3, 3, 2, 2, 0, 1, 2, 3)$ on the assignments, full marks on all labs, 19 out of 20 on the first midterm, 17 out of 20 on the second midterm, and 22 out of 25 on the final. What is Alice's overall score for the course (out of 100)?

2. (1 point) Review Chapter 0 of the textbook and answer the following questions.

   Let a, b, and n be integers with $a \leq b$ and $n \geq 1$. Let r be a real number.

   (a) Write a closed-form formula for $\displaystyle\sum_{i=a}^{b} i$.

   (b) Write a closed-form formula for $\displaystyle\sum_{i=0}^{n} r^i$.

   (c) Prove that $\displaystyle\sum_{i=1}^{n} \frac{1}{i} \leq 1 + \ln(n)$.

   (d) Prove by induction that $\displaystyle\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$.

(e) A bag contains seven cards numbered 1 through 7. The cards are drawn one at a time, uniformly at random without replacement. Drawing stops as soon as 2, 3, or 4 is drawn. What is the probability that the last card drawn is 2 or 4?

3. (1 point) A nonnegative integer can be written in binary (base two) as a sequence of bits. The following bitwise operators act on integers by viewing them as sequences of bits.

- AND (`&`): For each position, the output bit is 1 if and only if both input bits are 1.
  Example: `12 & 10 = 8`, since $(1100)_2$ `&` $(1010)_2$ `=` $(1000)_2$.
- OR (`|`): For each position, the output bit is 1 if and only if at least one input bit is 1.
  Example: `12 | 10 = 14`, since $(1100)_2$ `|` $(1010)_2$ `=` $(1110)_2$.
- Exclusive OR, XOR (`^`): For each position, the output bit is 1 if and only if the two input bits are different.
  Example: `12 ^ 10 = 6`, since $(1100)_2$ `^` $(1010)_2$ `=` $(0110)_2$.

The following bitwise operations shift the bits of the left operand:

- Right shift (`>>`): `x >> y` shifts the bits of x to the right by y positions, discarding the rightmost y bits.
  Example: `22 >> 2 = 5`, since $(10110)_2$ `>> 2 =` $(101)_2$.
- Left shift (`<<`): `x << y` shifts the bits of x to the left by y positions, inserting y zeroes.
  Example: `5 << 2 = 20`, since $(101)_2$ `<< 2 =` $(10100)_2$.

Be careful with operator precedence when using bitwise operations. You could try to predict the values of the following expressions before testing them in Python (this is not a question): `3 | 2 & 1`, `3 + 2 << 1`, `3 + 2 & 1`, and `3 & 2 << 1`.

For the following questions, let x and y be nonnegative integers.

(a) Evaluate the following expressions: `10 & 3`, `10 | 3`, `10 ^ 3`, `10 << 3`, and `10 >> 3`.

(b) Write an expression using only bitwise operators that evaluates to 1 if x is odd and 0 if x is even.

(c) Rewrite the following Python expressions using only bitwise operators: `x * (2 ** y)` and `x // (2 ** y)`.

(d) Rewrite the following Python expression using only bitwise operators and subtraction: `x % (2 ** y)`.