CSCI 0500: Data Structures, Algorithms, and Intractability (Fall 2025)

Assignment 0

Due at 11:59pm ET, Wednesday, Sep 24

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

   (a) Alice experiences network issues before the deadline and is unable to submit an assignment/workshop on Gradescope. Alice emails her solutions to the professor two minutes after the deadline, with file timestamps showing work completed before the deadline. What score will Alice receive?

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

   (c) Bob asks Alice to share her workshop/assignment solutions with him, promising to use them only for learning purposes. Alice agrees. Bob understands Alice's solution, writes his solution (which he can reproduce), and submits it. Is this a violation of course policy?

   (d) Suppose the course has 12 workshops (1 point each), 12 assignments (3 points each), and 10 labs (1.25 points each). Alice scores $(1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1)$ on workshops, $(3, 3, 3, 2, 2, 2, 1, 1, 0, 3, 3, 3)$ on assignments, full marks on all labs, 18 out of 20 on the midterm exam, and 25 out of 30 on the final exam. 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.

   (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 $\displaystyle\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$.

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 can test this in Python (this is not a question): `3 | 2 & 1`, `3 + 2 << 1`, `3 + 2 & 1`, `3 & 2 << 1`.

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

   (b) How can we decide if a nonnegative integer x is odd or even using bitwise operators?

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

   (d) Rewrite the following Python expressions using bitwise operators: `x % (2 ** y)`.