0. Basic Concepts

This problem checks your knowledge of some basic facts used in this course. Note: these do not cover all the needed background, but just some more fundamental ones. If you do not know how to answer some of these, review the Appendix of the textbook (which will be useful no matter what).

1. \( \lfloor 9.3 \rfloor = ? \) \( \lceil 9.3 \rceil = ? \)
2. \( \ln(e^x) = ? \) \( \log_{10}(1000) = ? \) \( \log_2(1024) = ? \)
3. \( \sum_{i=1}^{n} i = ? \)
4. \( \sum_{i=1}^{n} \frac{1}{i} = ? \)
5. What is approximately \( \sum_{i=1}^{n} \frac{1}{i} \)?
6. \( \sum_{i=1}^{n} \sum_{x=1}^{m} ix = ? \)
7. We define an unrooted tree as a undirected, acyclic graph. How many edges are there in an unrooted tree with \( n \) nodes?
8. Let \( x \) be a random binary variable (i.e. \( x \) can be 0 or 1), where the probability of \( x = 1 \) is 0.6. What is the expectation of \( x \) (i.e. \( E(x) \))?  
9. Let \( x \) and \( y \) be two random variables, whose expected values are 1 and 2 respectively. What is \( E(x + y) \)?
10. Let \( S \) be a set with \( n \) elements. How many distinct subsets does \( S \) have?

1. Matrix Multiplication

Given two \( n \) by \( n \) matrices \( A \) and \( B \), write an algorithm to compute the product of \( A \) and \( B \). Then, analyze your algorithm to find out how many additions and multiplications your algorithm will take. Here, treat each addition or multiplication of two whole integers as one operation.

2. The Pancake Problem

A stack of \( n \) pancakes is placed in front of you. You have a spatula which you can insert anywhere into the stack and flip over all the pancakes above the spatula. You want to arrange the pancakes in order of their diameter (they are perfectly round), and you want to use as few flips as possible. As an example suppose \( n = 6 \), and the pancakes are numbered 1 through 6 in order of their diameter with 1 the smallest and 6 the largest. Suppose the original order is 346215, and the left end of the sequence represents the top of the stack. In one flip I can get 643215 (by flipping the first three pancakes: 346), then in the next flip 512346, then 432156, then 123456, so four flips are enough in this case. Let \( F(N) \) be the worst case number of flips needed to arrange a stack of \( N \) pancakes. Find the most efficient (in a worst case sense) algorithm, that you can for this problem, where efficiency is measured by the worst case number of flips. Remember that your algorithm should work for pancakes with any order. To start you off you should easily be able to show that \( F(3) \) is at most 2n-2. Next, reduce that bound a little more if you can.