Homework 1: Complexity and ADTs

Data Structures

Q1 - Q4 are related to the ADT material. In the lecture, we talked about "outside the black box" and "inside the black box". The related discussion question will give you some practice with working "outside the black box". The assignment is intended to help you think about "inside the black box", by having you look at alternative representations. In Q1, you look at an alternative to binary. In Q2, you look at where stuff might be stored as a consequence of a selected implementation. In Q3 and Q4, you look at alternative data structures to leverage the fact that many locations in a standard array might always be empty. In Q6 and Q7 you look at how the time is affected by the underlying cost of the algorithm.

1. Show how nonnegative numbers can be represented in an imaginary ternary computer using trits (0,1,2) instead of bits (0,1). Why don't we do things this way?
2. If each element of an array RM, with 10 rows and 20 columns, stored in row major order, takes four bytes of space, where the first element of RM starts at 100, what is the address of RM[5][3] and RM[9][19]?

In questions 3 and 4, we suggest indexing the array from one, since matrices are normally indexed from (1,1). After you find your formula, then you can adjust it to index the array from zero.

1. A lower triangular matrix is an nxn array in which has a[i][j] = = 0 if i<j. What is the maximum number of non zero elements? How can they be stored in memory sequentially? Find a formula k= f(i,j) to store location a[i][j] in k (you only want to store the nonzero elements). Do not write code. Code would work if you were manually converting the matrix from one form to the other all at once, but you need the formula to convert otherwise.
2. A tridiagonal matrix is an nxn array in which has a[i][j] = = 0 if |i-j| > 1. What is the maximum number of non zero elements? How can they be stored in memory sequentially? Find a formula k= f(i,j) to store location a[i][j] in k, when |i-j| <=1 (you only want to store the nonzero elements). Do not write code. Code would work if you were manually converting the matrix from one form to the other all at once, but you need the formula to convert otherwise.
3. Consider two functions **f(n)=an2**, and **g(n)=bnlgn**. For what value of **n** do they intersect, and at which value(s) of the constants **a** and **b**? Pick some values of **a** and **b** and then find **n**. Experiment with different sets of values for **a and b**.  What trends do you observe? **an2** and **bnlgn** are terms that might come from an expression representing the amount of work done by a piece of code. Looking for where they are equal will help you decide which part is dominant. *We suggest solving this problem empirically, rather than mathematically, i.e. draw graph of the functions.* Remember that **a** and **b** are constants, but may not necessarily be integers.

For Q6 and Q7 just worry about time, not space. Feel free to leave your response in exponential form.

1. What is the maximum size of a problem that can be solved in one hour if the algorithm takes lg n microseconds?
2. What is the maximum size of a problem that can be solved in one hour if the algorithm takes n3 microseconds?