March 24, 2020 Name____Nigel Tan___________________

Computer Science 581 – Exam 2

1. Depth-first search.
    a. What is a back edge, can they exist, and if so how are they useful?
        a. A back edge is an edge connecting a vertex to one of its “ancestors” in the DFS tree. They exist for DFS and are useful for detecting cycles.
    b. What is a cross edge, can they exist, and if so how are they useful?
        a. A cross edge connects a vertex to one of its “sibling” vertices on the same level. They don’t exist for DFS unless it is a directed graph. They don’t aren’t particularly useful in a DFS tree.

2. Network flow.
    a. Describe how you would use it to search for a 5x5 Boolean matrix whose row sums are (2,2,3,3,4) and whose column sums are (1,2,3,4,4).
        a. I would use a series of bipartite matchings to verify whether a given matrix satisfies the row and col sum constraints.
        b. There would be 10 matchings total. Each constructed to model the sum of each row/column.
            i. Example for row 1 there is a source and sink vertex s and t respectively. There is a vertex R representing row 1 on the left side of the matching and 5 vertices on the right side corresponding to each potential column position that are connected to the sink. We then place an edge connecting R to a vertex x if the value for the matrix at (1,x) has a 1.
        c. For each bi-partite graph we see if the max flow or matching is equal to the desired row/col sum.
        d. Thus we have a way of verifying each row/col sum.
        e. I would then brute-force search for matrices with some heurstics based on the given row/col sums and use the network flow to test the solution.
    b. Show such a matrix or explain why no such matrix exists.

        1 0 1 0 0
        0 0 0 1 1
        0 0 1 1 1
        0 1 0 1 1
        0 1 1 1 1

3. Linear programming.
    a. What constitutes a proper problem formulation?
        a. Assuming we want standard form. We have an objective function to maximize. A group of linear inequalities as constraints. And non negativitiy constraints for all variables.

b. Describe the Simplex algorithm.
    a. Change the problem into its standard form, then its slack form.
    b. In the objective function choose the non basic variable that can contribute the most (has the largest positive coefficient).
    c. Choose the constraint with the tightest bound on the chosen non basic variable based on the basic feasible solution and make it the basic variable.
    d. Replace all instances of the new basic variable with the new equation for all the constraints and objective function.
    e. Repeat from step (b) until we cannot increase the objective function any more. The basic solution is now the optimal solution.

4. The FFT.
    a. How is it related to the notion of power spectrum?
        a. The FFT changes a general function from it domain to a frequency domain by representing it as the sum of frequencies.
    b. Exactly where in its derivation does mathematical symmetry enable us to multiply two polynomials of degree *n* in *O(n log n)* time? Be specific and concise.
        a. During the point-wise evaluation of polynomials A and B along with the interpolation of the pointwise polynomial C.
        b. A(x) = A_e(x^2) + xA_o(x^2) (A_e: even, A_o: odd)
        c. By exploiting the symmetry of the roots of unity we only need to evaluate/interpolate at n/2 roots of unity for both A_e and A_o instead of n roots. Thus both operations are in O(n log n)

5. Number-theoretic algorithms. Consider an RSA crypto system with *n*=91 and *E*=5.
    a. Encode the messages 2 and 3.
        a. 2 => 2^E mod n = 2^5 mod 91 = 32 mod 91 = 32
        b. 3 => 3^5 mod 91 = 243 mod 91 = 61
    b. Determine a suitable value for *D*.
        a. 91 = pq, 91 = 7*13, phi(91) = (p-1)(q-1) = (6*12) = 72
        b. d = e^{-1} mod phi(n)
            i. d = 29, 5*29 mod 72 = 145 mod 72 = 1

6. Complexity Theory. A vertex cover of a graph *G* is a set *S* of vertices such that every edge in *G* has at least one endpoint in *S*.
    a. Define the decision, search and optimization versions of the vertex cover problem.
        a. Decision: Does the graph have a vertex cover of a given size s?
        b. Search: Find the smallest vertex cover.
        c. Optimization: Find the size of the smallest vertex cover for G.
    b. Precisely what does it mean to say that the vertex cover problem is *NP*-complete?

a. The vertex cover problem is in NP and is at least as hard as all other problems in NP.

This exam is closed book. Simply insert your answers into this document and return it to Professor Langston (cc to Rachel) at the appointed time. Stay true and good luck!