Math 409 – Homework 1

Due on Thursday, April 4

You are welcome to talk with other students in the class about problems but should write up solutions on your own. Solutions can be handwritten or typed but need to be legible and submitted via Gradescope by the end of the day on Thursday. Late submissions can be turned in until the end of Sunday (see the syllabus for late policy). You should justify all your answers in order to receive full credit.

Problem 1. Consider the following algorithm with mystery subroutines:

> MYSTERY ALGORITHM
> **Input**: a graph G with n vertices
> **Output**: a number depending on G
> (1) For every $i = 1, \dots, n$, do,
>   (2) for every $j = 1, \dots, n$ do,
>   (3) compute $A(i, j)$.
> (4) For every $k = 1, \dots, n$
>   (5) for every $S \subseteq \{1, \dots, n\}$,
>   (6) compute $B(k, S)$.
> (7) For every $U \subseteq \{1, \dots, n\}$ with $|U| \leq 5$,
>   (8) compute $C(U)$.
> (9) Return $A(n, n) + B(n, \{1, \dots, n\}) + \min_{|U|=5} C(U)$

The running times of computing $A(i, j)$ in line (3), $B(k, S)$ in line (6), and $C(U)$ in line (8) are $O(n^2)$, $O(n)$ and $O(1)$, respectively, given the previously computed entries.

What is the running time of this algorithm? Give your answer in the form $O(f(n))$ for some function $f(n)$, simplified as much as possible, and justify your answer.

Problem 2. Let $T = (V, E)$ be tree with $n = |V| \geq 2$ vertices.

(a) Show that $\sum_{v \in V} \deg(v) = 2|E|$. (This will hold for arbitrary graphs, not just trees!)

(b) Show that T has some vertex of degree one (also called a *leaf*).

(c) Use (b) to prove by induction that T has exactly $|E| = n - 1$ edges.
Remark: This will also follow from a different proof in lecture, but you should give your own proof by induction here.

(d) Show that T has at most $n/2$ vertices with degree 3 or higher.

Remark: The solutions to (a) and (b) are unrelated. Both are useful for (d).

Problem 3. In the lecture, we saw an algorithm to compute the minimum cost of a TSP tour on the complete graph K_n in time $O(2^n n^3)$ using dynamic programming. Consider the following variant of this problem where $3 \leq m \leq n$ is a parameter:

> INPUT: $n \in \mathbb{N}$ with $m \leq n$ and edges costs $c_{ij} \in \mathbb{R}_{\geq 0}$ for $1 \leq i < j \leq n$
>
> GOAL: Find the minimum cost of an m-cycle in K_n.

Give an algorithm (based on the dynamic program for TSP) that solves the problem. (You should give the algorithm and argue that it successfully solves the problem.) The running time of the straightforward solution is the is the minimum of $O(n^3 2^n)$ and $O(n^3 n^m)$ — the second is better when m is much smaller than n.