Fall 2020: CSE 221 | **Discrete Mathematics** | **Akdeniz University**

Tuesday 29/01/2021 | Final Exam | Duration: 90 minutes

Name: | Student No:

P1 [5+10=15 points] Pigeonhole Principle

a) Suppose that you have an array of X numbers where X is your parameter for P1a. What is the largest number n such that you are guaranteed to have either an increasing or a decreasing subsequence of length n in your array?

b) How many ordered pairs of integers (a, b) are needed to guarantee that there are two ordered pairs (a_1, b_1) and (a_2, b_2) such that $a_1 \equiv a_2 \mod A$ and $b_1 \equiv b_2 \mod B$ where A and B are your parameters for P2b?

P2 [15 points] Rook Polynomials In the following table, delete(cross) all four squares having the numbers in your parameters for P2. Then, find the rook polynomial for the remaining table.

0	1	2
3	4	5
6	7	8

P3 [20 points] Generating Functions You will play a game with Elon Musk. He tells you that he'll give you 1 million dollars if you win the game. Rules are simple: you'll choose an integer n first. Then you'll roll a classical (six sided, numbered 1 to 6) fair die 8 times and if the sum is exactly n, you win.

a) What n will you choose to maximize your chance? (This is very easy.) (5pts)

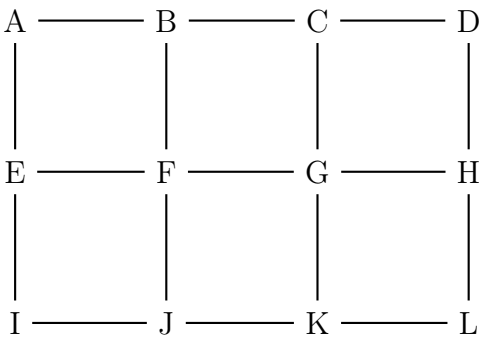
b) With the n you chose, what is your chance (probability) of winning? (15pts)

P4 [10+5=15 points] Bipartite Graphs

a) Draw a graph that is 3-regular, planar, and bipartite at the same time.

b) Can there be a 3-regular graph with 99 vertices? Why/why not? Prove your answer.

P5 [20 points] Graph Basics - MST First, write your personal parameters for this question in writing direction (left to right, top to bottom) in the given graph. Then, apply **Prim's algorithm** starting from node F to find a minimum spanning tree. Starting with F, write the vertices in order they are covered (visited). Also indicate the edges on the graph.



P6 [20 (10*2) points] Graph Programming Suppose that you already have a graph class and a method that returns the shortest path between two given vertices. (Assume all edges have weight 1.) Your task is to add two new methods to this class, getEccentricity() and getDiameter(). Fill these methods below. Recall that eccentricity of a vertex v is the distance to the farthest vertex to v and diameter is the distance between farthest vertices in a graph.

```
public class Graph {
    ArrayList<Vertex> vertices;
    public Graph() { /*Already written*/ }
    ...
    public int shortestPath(Vertex v, Vertex w) { /*Already written*/ }
    public int getEccentricity(Vertex v) { //Fill this method




    }
    public int getDiameter() { //Fill this method




    }
}
```