CS 591: Computational Game Theory	**Spring 2018**
Problem Set 2	
Lecturer: Prof. Peter Chin	*Due: Feb 21, 2018*

◇ Please email your written problems (must be typed), code and report to `kenzhou@bu.edu` by 23:59PM on the due date.

◇ Late policy: there will be a penalty of 10% per day, up to three days late. After that no credit will be given.

1. **(60 points) Written Problems**

(a) (15 points) Read the following three sections: 2:3-5; 5:1-2; 6:1 of the Osborne, and for each section, summarize in a paragraph of what your learned.

(b) (15 points) Find the g-function of the following games:

  i. the subtraction game with subtraction set $S = \{1, 3, 4\}$

  ii. one-pile game with the rule that you may remove at most half the chips. Of course, you must remove at least one, so the terminal positions are 0 and 1.

  iii. one-pile game with the rule that you may remove c chips from a pile of n chips if and only if c is a divisor of n, including 1 and n. For example, from a pile of 12 chips, you may remove 1, 2, 3, 4, 6, or 12 chips. The only terminal position is 0.

(c) (15 points) Subtraction games of problem 3:(a) - (c) of Problem Set 1.

(d) (15 points) Suppose you have 3 piles of (100, 200, 300) chips and play (b).1 for the first pile (the pile of 100), (b).2 for the second pile (the pile of 200), (b).3 for the third pile (the pile of 300). Should you go first or second? Justify your answer by computing the g -function of this game.

2. **(60 points) Programming**

(a) (30 points) **g-function calculator:** Write a program that computes g function of a combinatorial game. You can assume that the directed graph that represents the game has at least 1 terminal position. You can assume that the graph is connected and also assume that it can be represented by two text files such as `nodes.txt`, which contains all the nodes in the graph where each line contains the id of a node; and `edges.txt`, which contains all the edges in the graph where each line represents a directed edge, such as

source-node-id   desitnation-node-id

(b) (30 points) A generalization of Nim into a Take-and-Break Game is as follows: Suppose that each player at his turn is allowed (1) to remove any number of chips from one pile as in nim, or (2) to split one pile containing at least two chips into two non-empty piles (no chips are removed). Clearly the g function for the one-pile game satisfies $g(0) = 0$

and $g(1) = 1$. The followers of 2 are 0, 1 and (1, 1), with respective g values of 0, 1, and $1 \oplus 1 = 0$. Hence, $g(2) = 2$. The followers of 3 are 0, 1, 2, and (1, 2), with g values 0, 1, 2, and $1 \oplus 2 = 3$. Hence, $g(3) = 4$. Continuing in this manner, find the general form of g by computing the next 97 values of g up to $g(100)$ using your program from part (a). You may need to modify your program from part a).