# CS 677 - Analysis of Algorithm
# Fall Semester
# Homework 6 - Code Description

Dzung Bui

Due date: November 14, 2019

In homework 6, the program need to return the maximum number of events could be seen if the last event MUST be seen. It contain only one main function:
*bool maxS(int S[], int L[], int C[], int n)*: The function's inputs are three arrays and the number of event $n$. The first array $S$ are used to store the maximum number of events can be seen if the event with the same array index is selected. The second array $L$ is used to store the previous event should be seen if the event with array-index must be seen. The third array $C$ stores the coordinate of the events.

To fill out the two arrays $S$ and $L$, I used from bottom-up approach, start from event "0" and move to the next. For event whose index is large than "1", I determined which previous events can be seen by checking condition *time-to-move* ≥ *difference in coordinate*. If there are multiple event satisfied, the algorithm would select the one which can provide the larger number of event can be seen.

The outcome of the function:

- print out the list of event and its coordinate
- Maximum number of events could be watched if the last event (event 9) MUST be seen.
- An array of maximum number of events could be seen if the event with array-index MUST be seen.
- An List consists of the previous event should be seen for a particular event.

To comply my code, use command *g++ -o hw6 dynamicProgramming.cpp*, and run the executed file by command *./hw6* in the terminal.