# AI 2016 Homework 2

2016/03/18

# Pacman Project

- http://ai.berkeley.edu/project_overview.html

- Project 0: Unix/Python/Autograder Tutorial
  - These are prerequisites; people should make sure they are competent of doing Project 0.

- Project 1: Search in Pacman
  - Fully observable, single agent and deterministic

# HW2 Requirements

- Write correct uniform-cost search and A* search
  - Passing autograder does not mean there are no bugs.


- Pass all autograder test cases of q3 and q4
  - E.g. $python autograder.py -q q3

# Submission

- Upload a zip file to course website.
  - <student_id>.zip (e.g. r04944021.zip)
  - Which includes search.py and README

- Due at 23:55, Mar. 26, 2016
  - Delaying less than one week: 20% off
  - No more than one week delay

# FAQ – I couldn't find a submit button

- You should be enrolled in the course

- Navigate to "Artificial Intelligence (2016 Spring)" on course website
  - https://course.agent.csie.ntu.edu.tw/
  - Enroll yourself

# FAQ – I couldn't run pacman.py

- ImportError: No module named _tkinter
  - http://tkinter.unpythonic.net/wiki/How_to_install_Tkinter

- Xorg crash
  - https://bugs.launchpad.net/ubuntu/+source/xorg/+bug/1509507

- Use Print "foo" for python2
  - Use Print("foo") for python3

# FAQ – Was the BFS algorithm in course slides wrong

- Autograder expects you to check if a goal state is reached after popping from a search frontier

- The algorithm on course website checks if a child state is goal while expanding a state
  - Correct and faster on average
  - But autograder fails it

# FAQ – My code went haywire with some test cases

- You might be expecting a state is a tuple of coordinates or an action is a string like "south"

- Use functions provided by the problem instead
  - E.g. problem.getSuccessors()

# FAQ – Tiebreaker when popping from a search frontier

- The search path must be deterministic for autograder

- When expanding a state, add child states into a search frontier in the order (or reverse) provided by problem.getSuccessors()

- Use the priority queue defined in util.py