15-110 recitation 10

Recap

- graphs, representation, searching, BFS/DFS
- Trees
- Tractability, TSP, P v. NP
- Lists- iterating, 2D lists, aliasing, destructive methods
- Recursion
- Big O and Complexity
- Dictionaries

Reminders!

HW 5 check-in due Monday!

Exam 2 tomorrow

Problems

P v. NP Fast Facts

Problem

Some intractable problems are in P [T/F]
All problems in P are in NP [T/F]
If we find a tractable solution to exam scheduling, then P = NP [T/F]
The Travelling Salesman Problem is NP-Complete. [T/F]

getColumn

Problem

Write the function **getCol(L, c)** that takes a 2d list L and a column c and returns a 1d list of the given column of L, or None if no such column exists. For example, if L = [[1, 2, 3], [4, 5, 6]], then getCol(L, 0) returns [1, 4] and getCol(L, 2) returns [3, 6], and getCol(L, 3) returns None.

```
def getColumn (L):
```

list alias code trace

Problem

```
def ct1(L):
      A = [1,2,3]
      B = L
```

```
C = A
C[0] = 3
A = B
A[1] = 5
print (A)
print(B)
print(C)
```

ct1([9,8,7,6]):

listSum

Learning Objectives

- get more comfortable with trees & graphs and their dict representation

Write a function to sum all the elements of a list called listSum.

Problem

listSum([2, 3, 4, 5]) -> 14

```
def listSum (L):
```

graphs

Problem

Given the following dictionary, draw the graph it represents. (Bonus 1: Write out the adjacency matrix that represents the same graph.) (Bonus 2: Starting at `Alpha`, trace a DFS and a BFS for `Demeter`.)

```
graph = { "Alpha": ["Delta", "Apollo", "Artemis"],
          "Delta": ["Alpha", "Demeter", "Dionysus"],
          "Apollo": ["Alpha"],
          "Artemis": ["Alpha"],
          "Demeter": ["Delta"],
          "Dionysus": ["Delta"]}
```