Attendance:
tiny.cc/event-attendance

App Dev League

Lesson Overview

1. Review of last session
2. Recursion
3. After this we'll take a break
4. Graphs

1 Review

Basic Python

- Loops
  - For loops
  - While loops
- Conditionals
- Data Structures
- Time Complexity
- Searching and Sorting

2 Recursion

What is Recursion?

- Dividing a problem into a subproblem
- Calling a function inside the same function
- Returning a value when the base case is reached

Calculate Factorial Using Recursion

- Let’s define fact(n) as n!
- How do we solve for fact(n) using recursion?
- fact(n) = n*fact(n-1)
- Base case: n = 1, 1! = 1

Calculate Fibonacci Sequence

- Let’s define f(n) as the nth fibonacci number
- Each number in the sequence is the sum of the two numbers that precede it
  - 0, 1, 1, 2, 3, 5, 8, …
- f(1) = 0, f(2) = 1
- f(n) = f(n-1)+f(n-2)

Exercise: Calculate arithmetic series

- Problem: Given an integer n, find f(n) = 1+2+3+...+n using recursion
- Hint 1: Create a base/terminating case (What is f(1)?)
- Hint 2: Come up with an equation for f(n) that includes the term f(n-1)

Exercise 2: Calculate sum of digits

- Problem: Given an integer n, calculate the sum of the digits of n
  - Ex: If n = 231, the sum of the digits of n is 2+3+1=6
- Hint 1: Define the function - f(n) is the sum of the digits of n

- Hint 2: Come up with an equation for f(n) that includes f(n/10)

- Hint 3: What's the base case?

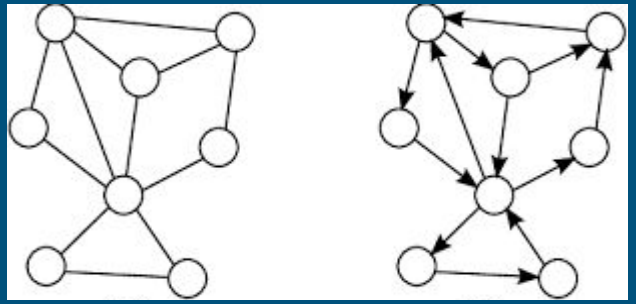
Examples of how to use recursion

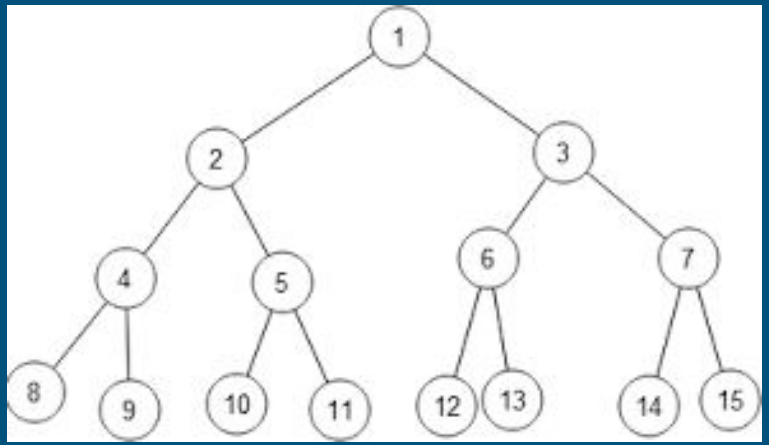
- Generating Subsets
  - You can either have the element or not have it
  - Dependent on the subsets without it
- Generating Permutations
  - Run through all possible cases of all orders of the array
  - An array with n elements depends on the number of permutations with n-1 elements

3 Graphs

What are Graphs?

- A network between points and lines
- There are two types of graphs
  - Directed and Undirected
- Sparse and dense graphs
- Trees





Graph Representation

- Adjacency Matrix
  - A n by n grid that tells you if a there is an edge between two nodes

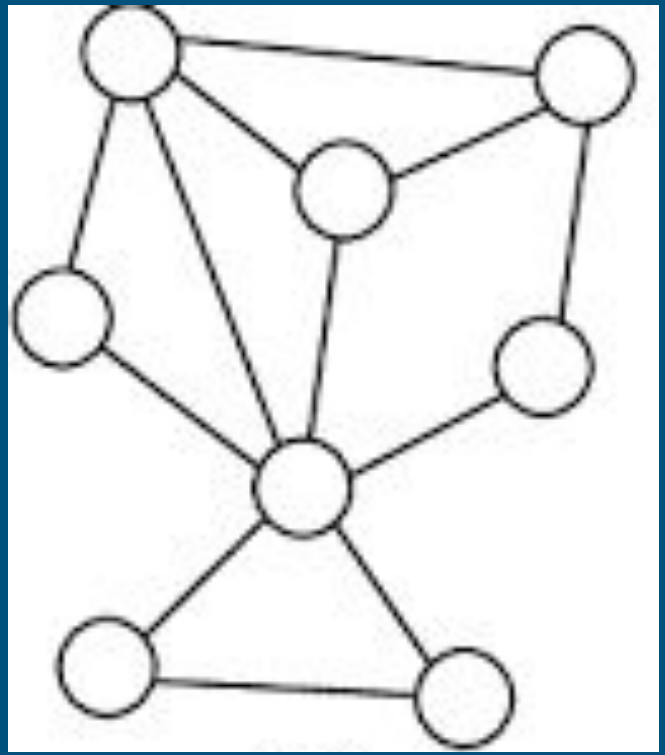
```
adj_mat=[[False]*(n+1)]*(n+1)
```

- Adjacency List
  - A list of arrays that tells use which nodes have an edge with another node

```
adj_list=[[]]*(n+1)
```

Graph Traversal or DFS

1. Start at one node of the graph
2. Visit one of its neighbors
3. Do the same for that node
4. Continue until all nodes are visited



Graph Problems

- Milk Factory
- Grass Planting

Problems in General

1) Maximum Distance
2) Teleportation
3) Triangles
4) Just Stalling

Content Review

- ➔ Had a review on searching and sorting
- ➔ Learned how to solve recursive problems
- ➔ Learned what graphs are
- ➔ Had fun!!

4 Conclusion

Course Review

- ➔ Day 1: Intro to Bootstrap
- ➔ Day 2: Intro to Flask
- ➔ Day 3: AI and Neural Networks
- ➔ Day 4: CNNs
- ➔ Day 5: Flask and Neural Net Project
- ➔ Day 6: Project Cont.
- ➔ Day 7: Algorithms in Python
- ➔ Day 8: Algorithms in Python

Feedback and Certificates

http://tiny.cc/advancedCSfeedback

THANKS!

ANY QUESTIONS?

You can find more info @

- https://www.appdevleague.org
- https://linktr.ee/AppDevLeague