CompSci 260P: Week 4

Midterm Preparation

Ryuto Kitagawa

University of California, Irvine

Before We Begin

- Hi, my name is Ryuto Kitagawa, and I'm one of your TA's :)
- Office Hours: Mondays from 2:00 - 5:00pm at ICS 458F
- Next week Thursday is Midterm

Ring Problem

- Suppose you are given a ring which looks like the one shown below
- Each step, you may turn the ring one letter or press the button
- Let K be a word you want to spell
  - What is the minimum number of steps necessary to spell K?


center

Ring Problem: Solution

- Observation: Can we represent this as a graph?
- What would be a natural start to our graph?
  - Let start of string be the first node in our graph
- What would the adjacent nodes be?
  - The next letter in K!
  - Note: You could make it the adjacent letters in the ring, which is a valid solution, but less efficient
- Should the graph be weighted or unweighted?
  - Weighted by number of steps it takes to enter that letter
- How do we find the answer in this graph?
  - Use Dijkstra to find the shortest path!

Recursion Fundamentals

- Recursion is the core of dynamic programming
- Make the problem smaller in some way
  - What that means depends on the problem!
- Smaller problem is input to your algorithm
  - Assume output here is correct!
- These two key points should shape what your recursive algorithm looks like
- **Don't forget the base case!**

Recursion Fundamentals

Dynamic Programming Fundamentals

- Dynamic Programming is the natural extension of a recursive algorithm
- Do not redo computations!
  - Find repetitive computations and stop them
  - Find the order of dependencies to linearize the algorithm
- This takes time to understand!


big

Company Party

- Suppose you are organizing a company party
- The corporation has a hierarchy tree
  - An employee has some subordinates, who have some subordinates, etc.
  - Creates a hierarchy tree
- If an employee is invited, their immediate supervisor cannot be invited
- Each employee i has a value v_i for how much value they bring to the party
  - Note: Assume the people organizing this party are jerks
- Produce an algorithm which maximizes the value of the party

Company Party

Company Party: Recursive Solution

- Consider the tree containing the hierarchy of all employees
- Start with the root:
  - Can you make a decision on this employee?
  - Invite, or don't invite!
- What about the root's children?
  - If we invited the root, then we can *not* invite them!
- Pass valid subtrees into the recursive function!

Company Party: Iterative Solution

- How do we linearize the algorithm?
  - Solve the base case first!
  - Then solve the next dependent parts
- Compute the function starting with the leaves
  - Solve all nodes at the lowest height
  - Then solve nodes one level above
  - Keep going all the way to the top!