CS 170 Section 6
Dynamic Programming

Owen Jow | owenjow@berkeley.edu

Agenda

- Dynamic programming
    - Shortest paths
    - String shuffling

Dynamic Programming

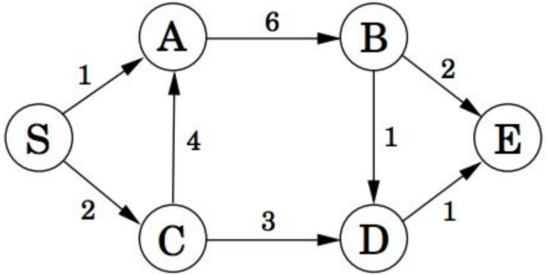
Dynamic Programming

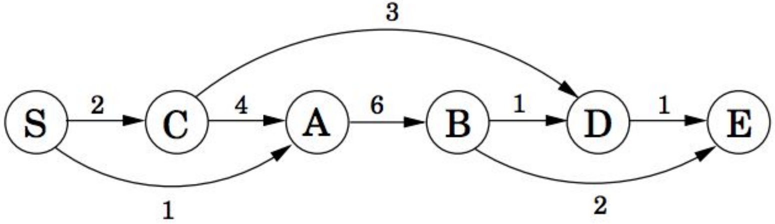
- Solve subproblems, then use the subproblems to solve bigger and bigger problems
- Can be implemented as recursion with memoization, or iteration that stores results in a table
- Key question: *what are the subproblems?*
  - should be able to solve subproblems given the results of “smaller” subproblems

You’ve probably seen dynamic programming many times in many forms. It just might not have been called “dynamic programming.”

Shortest Paths

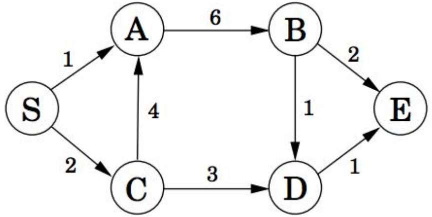
- Our goal is to find the minimum-length path from the start S to the end E. *What are the subproblems?*

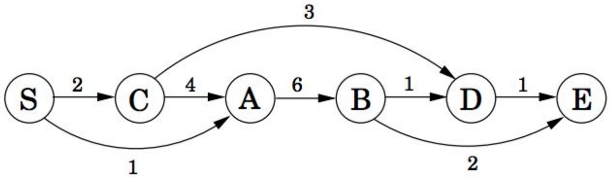




(the right graph is just the linearization of the left graph)

Shortest Paths





- **Subproblems**: what’s the shortest path to the predecessors of E? What’s the shortest path to the predecessors of those predecessors?... all the way back to *what’s the shortest path to S?*
- The bigger problems depend on the answers to the smaller problems (just add an edge)

Note: this formulation only works if the graph is directed and acyclic.

String Shuffling

.......what are the subproblems?.......
- your subconscious

- x, y, and z are strings
- Can we obtain z by interleaving characters from x and y (such that the characters for each of x and y still appear in their original order)?

For example, let x = “hello” and y = “WORLD”.

Answer: Yes	**Answer: No**
z = “hWeOlRlLoD”	z = “hWeOlRlLoDextra”
z = “helWOlRLDo”	z = “helWOlRLD”

String Shuffling

- **Subproblems:** can we interleave x[0:i] and y[0:j] to make z[0:i+j]?
- Formally, we can define S(i, j) to be True if so, and False if not

Then we can relate subproblems to smaller subproblems as
S(i, j) = True if {x[i] = z[i + j] *and* S(i - 1, j) = True} *or* {y[i] = z[i + j] *and* S(i, j - 1) = True}

Our base cases:

- S(i, 0) = True for the first i characters of x that are also the first i characters of z
- S(0, j) = True for the first j characters of y that are also the first j characters of z