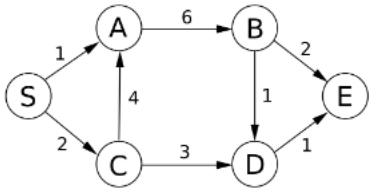
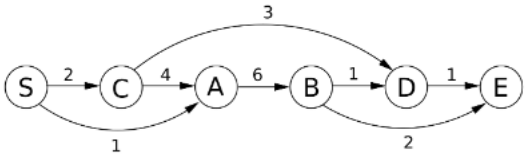
CS 170 DIS 06

Released on 2018-02-28

1 Shortest Paths with Dynamic Programming!





Write a dynamic programming algorithm to determine the minimum length path from the start (S) to the end (E).

1. What are the subproblems you need to solve to find the minimum length from S to E?

2. Can you define the optimal solution to any subproblem as a function of the solution to its own subproblems? What does the dependency graph look like? The dependency graph is a graph in which each subproblem becomes a vertex, and edge (u, v) denotes that subproblem u requires the solution to subproblem v in order to be solved. What about the given graph makes this particular DP solution convenient here?

3. If we were to proceed iteratively, what would be the best order to solve these subproblems in?

4. How many subproblems are there, and how long does it take to solve each? What is the overall running time of this algorithm?

5. Run your algorithm on the graph. What is the shortest graph?

2 String Shuffling

Let x, y, and z be strings. We want to know if z can be obtained only from x and y by interleaving the characters from x and y such that the characters in x appear in order and the characters in y appear in order. For example, if $x =$ **efficient** and $y =$ **ALGORITHM**, then it is true for $z =$ **effALGiORciIenTHMt**, but false for $z =$ **efficientALGORITHMextraCHARS** (miscellaneous characters), $z =$ **effALGORITHMicien** (missing the final t), and $z =$ **awoefnawolnef** (obviously wrong). How can we answer this query efficiently? Your answer much be able to efficiently deal with strings such as $x =$ **aaaaaaaaaab** and $y =$ **aaaaaaaac**

1. Design an efficient algorithm to solve the above problem and state its runtime.

2. Consider an iterative implementation of our DP algorithm in part (a). Naively if we want to keep track of every solved sub-problem, this requires $O(m * n)$ space (double check to see if you understand why this is the case). How can we reduce the amount of space our algorithm uses?