TCSS 343 - Week 7 - Monday

Jake McKenzie

March 3, 2019

Dynamic Programming

"People say nothing is impossible, but I do nothing every day."
...
Winnie the Poo

"It is not enough to be in the right place at the right time. You should also have an open mind at the right time."
...
Paul Erdős

"Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less."
...
Marie Curie

0. What are the two characteristics of a dynamic programming problem? (What kind of substructures and subproblems do you have?)

1. Is merge sort a candidate for dynamic programming? Why or why not?

2. Given a string S = ABAZDC and T = BACBAD compute the longest common subsequence between the two strings. For your convenience:

$$L(i,j) = \begin{cases} 0 & \text{if } i = 0 \vee j = 0 \\ \text{L[i-1,j-1]} + 1 & \text{if } i > 0, j > 0 \wedge a_i = b_j \\ \max\{\text{L[i,j-1],L[i-1,j]}\} & \text{if } i > 0, j > 0 \wedge a_i \neq b_j \end{cases}$$

3. Look back to your notes from lecture for the knapsack problem. Imagine you have a homework assignment with different segments labeled A through G. Each part has a "value" (in points) and a "size" (time in hours to complete). For example, say the values and times for our assignment are:

	A	B	C	D	E	F	G
value	7	9	5	12	14	6	12
time	3	4	2	6	7	3	5

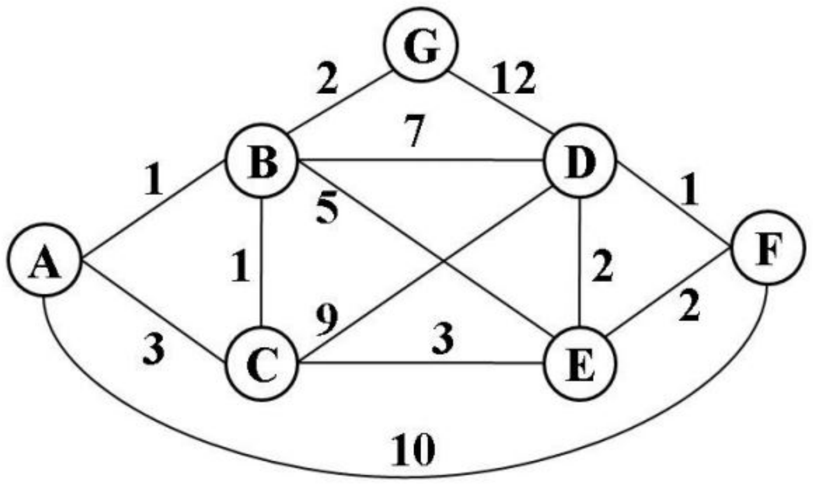
Say you have a total of 15 hours: which parts should you do? If there was partial credit that was proportional to the amount of work done (e.g., one hour spent on problem C earns you 2.5 points) then the best approach is to work on problems in order of points/hour (a greedy strategy). But, what if there is no partial credit? In that case, which parts should you do, and what is the best total value possible?

4. Let us say we have a set of rectangular blocks $\{1, 2, \ldots, n\}$, where each block has a length l_i, a width w_i, and height h_i. You want to stack blocks to get the maximum height. In order to stack block j on top of block i you require $l_j < l_i \wedge w_j < w_i$. Also there is a single orientation allowed for each block, where each side has a fixed north, south, east and west. Finish the reduction below:

$$RB(1, 2, \ldots, n) = \max_i \{h_i + RB(\qquad\qquad\qquad\qquad\qquad\qquad)\}$$

5. Let us now analyze the runtime. The first question you should always ask: How many subproblems do we have? Use the reduction to help you find what subproblems there are for this problem.

6. Consider the following undirected, weighted graph:



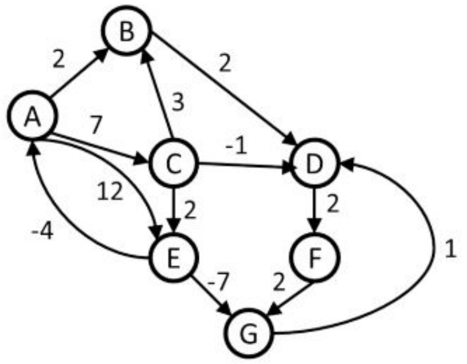
Step through Dijkstra's algorithm to calculate the single-source shortest paths from A to every other vertex. Show your steps in the table below. Cross out old values and write in new ones, from left to right within each cell, as the algorithm proceeds. Also list the vertices in the order which you marked them known. Finally, indicate the lowest-cast path from node A to node F.

Known vertices: _____ _____ _____ _____ _____ _____ _____
(in order marked known)

Vertex	Known	Cost	Path
A			
B			
C			
D			
E			
F			
G			

Lowest-cost path from A to F: ______________________________

7. Consider the following directed, weighted graph:



Even though the graph has negative weight edges, step through Dijkstra's algorithm to calculate *supposedly* shortest paths from A to every other vertex. Show your steps in the table below. Cross out old values and write in new ones, from left to right within each cell, as the algorithm proceeds. Also list the vertices in the order which you marked them known. Dijkstra's algorithm found the wrong path to some of the vertices. For just the vertices where the wrong path was computed, indicate both the path that was computed and the correct path.

Known vertices: ____ ____ ____ ____ ____ ____ ____
(in order marked known)

Vertex	Known	Distance	Path
A			
B			
C			
D			
E			
F			
G			

8. What single edge could be removed from the graph such that Dijkstra's algorithm would happen to compute correct answers for all vertices in the remaining graph?