The Traveler's Problem

Given an acyclic weighted directed graph G, two vertices s and t in G, and a fixed number B (the "budget"), find the shortest path (*i.e.*, fewest edges) in G from s to t, whose weight does not exceed B.

Figure 1 shows an instance of such a graph G. The least cost path has weight 7, but that is not the question. If $B = 10$, what is the shortest path? If $B = 12$, what is the shortest path?

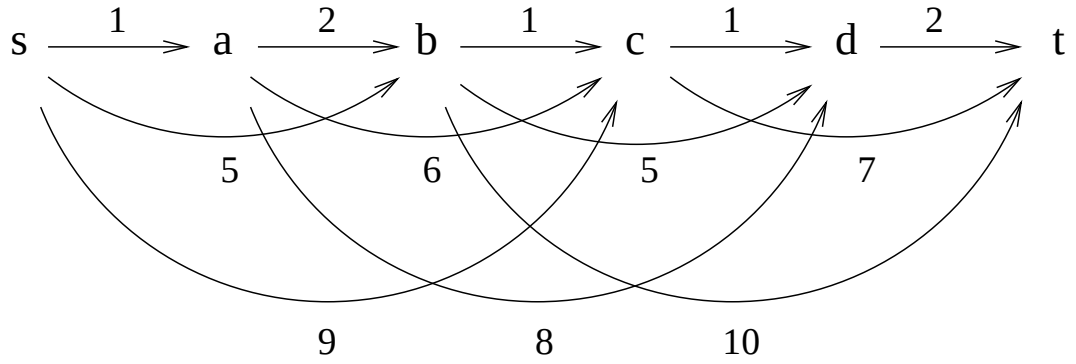

Figure 1: An Instance of the Traveler's Problem

Dynamic Programming Method

Draw a table where each row corresponds to a path length, and each column corresponds to a vertex. For any vertex x, the entry in row ℓ and column x is the least cost of any path of length ℓ from s to x. We also enter a back pointer, which is a vertex. If no such path exists, we can use the default entry ∞. The cost of our final path will be a number in the last column. We use the back pointers to recover the solution.

In the table below, I filled in one entry for you.

	s	a	b	c	d	t
0	0	∞	∞	∞	∞	∞
1						
2				$6, b$		
3						
4						
5						