Finish the Knapsack example from before

**Graphs with negative weights**

- Given a graph \( G = (V, E) \) with a weight function \( w: V \times V \rightarrow R \)
  - In other words, it could have negative weights
- Can we use Dijkstra’s algorithm to find shortest paths in a graph with negative edge weights?

**Graphs with negative weights**

- Why?
  - Is it the design of the algorithm?
  - Or that there are no shortest paths in graphs with negative edge weights?
- What is the shortest path distance between \( s \) and every other node in the graph?
Another example

• What about the shortest path between s and t here?

Negative cycles

Def. A negative cycle is a directed cycle such that the sum of its edge weights is negative.

Shortest paths and negative cycles

If some path from v to t contains a negative cycle, then there does not exist a cheapest path from v to t.

If G has no negative cycles, then there exists a cheapest path from v to t that is simple (and has ≤ n – 1 edges).

How can we solve the Shortest Path problem as a dynamic program

Let's think together
**Shortest-Paths** (V, E, c, 0)

**Foreach node v ∈ V**

\[ M[0, v] \leftarrow \infty. \]

\[ M[0, v] \leftarrow 0. \]

**For i = 1 to n - 1**

**Foreach node v ∈ V**

\[ M[i, v] \leftarrow M[i-1, v]. \]

**Foreach edge (v, w) ∈ E**

\[ M[i, v] \leftarrow \min \{ M[i, v], M[i-1, w] + c_{vw} \}. \]

---

**Bellman–Ford (V, E, c, r)**

**Foreach node v ∈ V**

\[ d(v) \leftarrow \infty. \]

\[ \text{successor}(v) \leftarrow \text{null}. \]

\[ d(r) \leftarrow 0. \]

**For i = 1 to n - 1**

**Foreach node w ∈ V**

If \( d(w) \) was updated in previous iteration

**Foreach edge (v, w) ∈ E**

If \( d(v) + c_{vw} \) \( < d(w) \)

\[ d(w) \leftarrow d(v) + c_{vw}. \]

\[ \text{successor}(w) \leftarrow v. \]

If no \( d(w) \) value changed in iteration i, STOP.