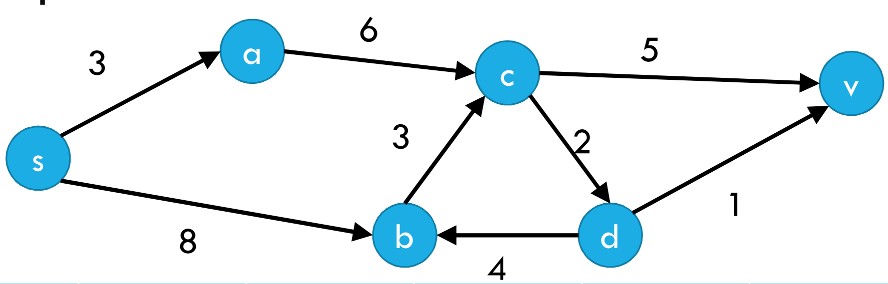
A recurrence

$$dist(v) = \begin{cases} 0 & \text{if } v \text{ is the source} \\ \min_{u:(u,v)\in E} \{dist(u) + weight(u,v)\} & \text{otherwise} \end{cases}$$

Our memoization structure can be the graph itself.

What's an evaluation order? (Remember we're in a DAG!)

Sample calculation



Vertex\\i	0	1	2	3	4	5
S	0	0	0	0	0	0
A	∞	3	3	3	3	3
B	∞	8	8	8	8	8
C	∞	∞	9	9	9	9
D	∞	∞	∞	11	11	11
V	∞	∞	∞	14	12	12

Pseudocode

```
Initialize source.dist[0]=0, u.dist[0]=∞ for others
for(i from 1 to ??)
   for(every vertex v) //what order?
      v.dist[i] = v.dist[i-1]
       for(each incoming edge (u,v))//hmmm
          if(u.dist[i-1]+weight(u,v)<v.dist[i])
             v.dist[i]=u.dist[i-1]+weight(u,v)
          endIf
      endFor
   endFor
endFor
```

$$dist(v,i) = \begin{cases} 0 & \text{if } i = 0 \text{ and } v \text{ is the source} \\ \infty & \text{if } i = 0 \text{ and } v \text{ is not the source} \\ \min\left\{ \min_{u:(u,v)\in E} \{dist(u,i-1)\} + w(u,v), dist(v,i-1) \right\} & \end{cases}$$

Takeaways

Some clever dynamic programming on graphs.

Which library to use (at least asymptotically)?

Need just one source?

Dijkstra's if no negative edge weights.

Bellman-Ford if negative edges.

Need all sources?

Flord-Warshall if negative edges or $m \approx n^2$

Repeated Dijkstra's otherwise

These are all asymptotics! For any "real-world" problem prefer running actual code to see which is faster.