16. Sea D un digrafo conexo que no tiene ciclos dirigidos, v el único vértice de D con grado de entrada 0[3] y $c\colon E(D) \to \mathbb{Z}$ una función de pesos.

a. Definir una función recursiva $d\colon V(D) \to \mathbb{Z}$ tal que $d(w)$ es el peso del camino mínimo de v a w para todo $w \in V(D)$. **Ayuda:** considerar que el camino mínimo de v a w se obtiene yendo de v hacia z y luego tomando la arista $z \to w$, para algún vecino de entrada z de w; notar que la función recursiva está bien definida porque D no tiene ciclos.

b. Diseñar un algoritmo de programación dinámica top-down para el problema de camino mínimo en digrafos sin ciclos y calcular su complejidad.

c. (Integrador y opcional) Diseñar un algoritmo de programación dinámica bottom-up para el problema. **Ayuda:** computar d de acuerdo a un orden topológico $v = v_1, \ldots, v_n$ donde $v_i \to v_j$ solo si $i < j$. Este orden se puede computar en $O(n + m)$ (guía 3).

a

$$d(w) = \begin{cases} 0 & \text{si } v = w \\ \min\{d(u) + c(uw) \text{ con } u \in N^{in}[w]\} & \text{c.c.} \end{cases}$$

b CaminoMinEnDAGs

1. Inicializo M vector de n posiciones en $\perp$

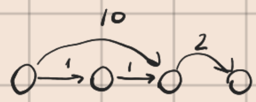
RecursiónTD(G, v, w)
  Return M[w]

RecursiónTDAux (G, v, w)
  Si v = w return 0
  Sino
    Si M[w] = $\perp$
      M[w] = min {RecursiónTD(G, v, u) + c(uw) / $\forall u \in N^{in}[w]$}
  Return M[w]

En cada paso recursivo exploramos para todo vértice su vecindario de entrada completo $\longrightarrow$ $O(n+m)$

(c) TD → vamos de w a v → "atrás para adelante"
BU → vamos de v a w → "hacia adelante"

Para el BU necesitamos el orden Topológico



Algoritmo BU

1. Orden Topológico de G desde v $O(n+m)$
2. Vértice actual = v.
3. $M[v] = 0$
4. Para $i=1,\dots,n$ $O(n+m)$
5. Para cada $u \in N^{out}(v_i)$
6. Si $M[u] = \perp$
7. $M[u] = c(v_i,u)$
8. Sino
9. $M[u] = \min\{M[u], M[v_i] + c(v_i,u)\}$
10. Return $M[w]$