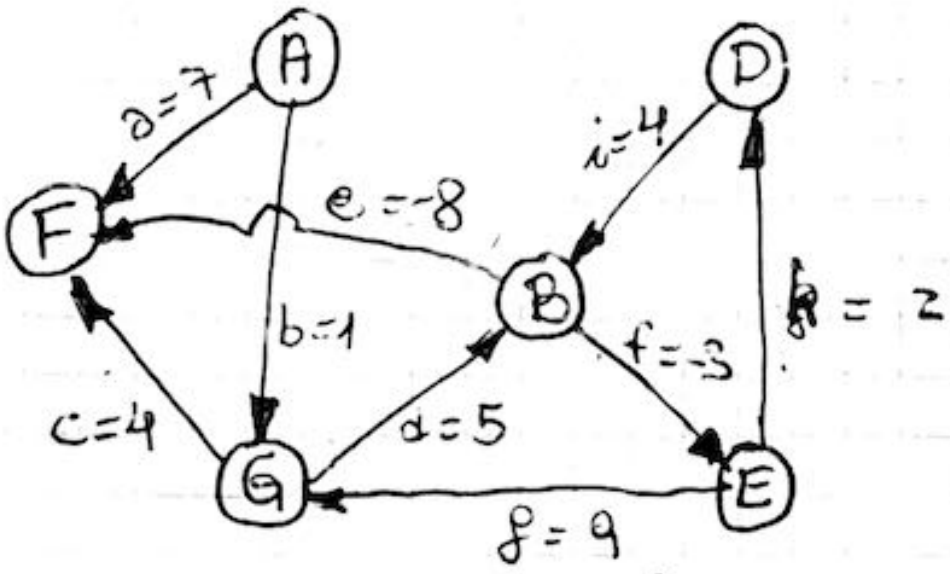
Ejercicio 1 1/2

CITERI IVÁN
100223



1) Creo un diccionario de distancias y uno de padres.

Padre	Distancias
A: None	A: 0
F: ~~A G B~~	B: ~~∞~~ 6
~~G~~: A	E: ~~∞~~ 3
B: G	F: ~~∞ 7 5~~ -2
E: B	G: ~~∞~~ 1
D: E	D: ~~∞~~ 5

2) Itero V veces. Tengo 6 vértices => itero 6 veces, ~~todas las aristas~~ cada arista.

3) Inicialmente las distancias desde A a los demás es ∞. Y, el padre de A es None porque es el origen.

4) Guardo ~~mal~~ etiqueta con letras minúsculas a los vértices para seguir un orden (uno en el apartado)

⑤ Empieza en la primera iteración (itera en orden alfabético PD: aclaro, no en el aportado, sino para hacer el seguimiento)

. Empieza desde a; a pesa 7, y el camino desde A hasta F es ∞, entonces mejora y lo cambio y el padre de F es A.

Sigo con b (A-G), pesa 1 y mejora a ∞ => lo cambio y el padre de G es A.

Sigo con c (G-F), pesa 4, A-G = 1 + G-F = 5 => mejora contra la distancia de F que es 7 ~~aportado~~, actualizo padre y distancia.

Sigo con d (G-B), la distancia de G es 1, y la de B ∞ => dist(G) + peso(d) = 6 => mejora cambio padre y distancia.

Sigo con e (B-F), pesa -8, ~~pesaba~~ dist(B) = 6 y dist(F) = 5, dist(B) + (-8) = -2 => ~~no~~ -2 < d(F) => cambio padre y distancia.

Sigo con f (B-E), ~~p~~ dist(B) = 6, dist(E) = ∞, dist(B) + peso(f) < dist(E) => actualizo padre y dist. => dist(E) = 3

Sigo con g (E→G), dist(E)=3, dist(G)=1, => => dist(E) + peso(g) > dist(G) => no hago nada

sigo con h (E→D), dist(E)= 3, dist(D)=∞ d(E) + peso(h) < dist(D) => actualizo padre y dist.

sigo con i (D→B), dist(D) = 5, dist(B) = 6, i = 4
=> dist(D) + i > dist(B) => no hago nada.
Ahora tercera la primera iteración
2° Iteración:
a) ~~(B → F)~~ (A → F), dist(A) = 0, dist(F) = -2, a = 7
=> d(A) + a = 7 > dist(F), no hago nada
b) (A → G), d(A) = 0, d(G) = 1, b = 1, no mayor, no hago nada.
c) (G → F), d(G) = 1, d(F) = 5, c = 4, no mayor.
d) (G → B), d(G) = 1, d(B) = 6, d = 5, no mayor.
e) (B → F), d(B) = 6, d(F) = 5, e = -8, no mayor
f) (B → E), d(B) = 6, d(E) - B, f = -3 no mayor
g) (E → G) d(E) = 3, d(G) - 1, g = 9 " "
h) (E → D) d(E) = 3, d(B) - 3, h = 2 " "
i) (D → B) d(D) = 5, d(B) = 6, i = 4 " "

desde la 3° iteración a la última no ha a habido cambios por el orden de aristas que elegí

Para este caso y para cualquier caso con grafos *~~cuyos costes~~ dirigido el cuál alguna arista ~~pueda~~ tenga peso negativo y quiera encontrar el camino mínimo desde un punto a otro no hay otra forma, por más que cueste el algoritmo es el que hay que utilizar.
Dijkstra es mejor en coste temporal pero no sirve para grafos dirigidos con aristas de peso negativo.
Además Bellman Ford encuentra ciclos negativos