::Dijkstra::

1. Se inicia un diccionario de padres y otro de distancias. También un heap de mínimos.
    - El diccionario de distancias se inicia con 0 para el nodo desde el cual se quieren obtener los caminos mínimos y distancia infinito para el resto.
2. Se agarra el nodo desde el cual se quieren obtener los caminos mínimos para comenzar el recorrido. Se lo agrega a padres (su padre es None). Se lo encola en el heap con distancia 0.
3. Si el heap no está vacío, se saca un nodo del heap.
4. Del nodo que se desencoló [se lo llamará A por practicidad] se revizan sus adyacentes, y por cada uno [se lo llamará B por practicidad]:

    ¿dist[B] > dist[A] + peso(A, B)?

    a. Si la rta es que sí, entonces:
        - Se actualiza el diccionario de distancias, en donde dist[B] = dist[A] + peso(A, B).
        - Se actualiza el diccionario de padres en donde el padre de B es A.
        - Se encola a B en el heap con su nueva distancia.

    b. Si la rta es que no entonces no se hace nada y se sigue con el algoritmo.
5. Se vuelve al paso 3 hasta que el heap no esté vacío, si lo está entonces se terminó el algoritmo.

::Complejiad::

(V: # vértices, E: # aristas)

Inicializar el resto de diccionarios → O(1)

Inicializar el diccionario de dist → O(V)

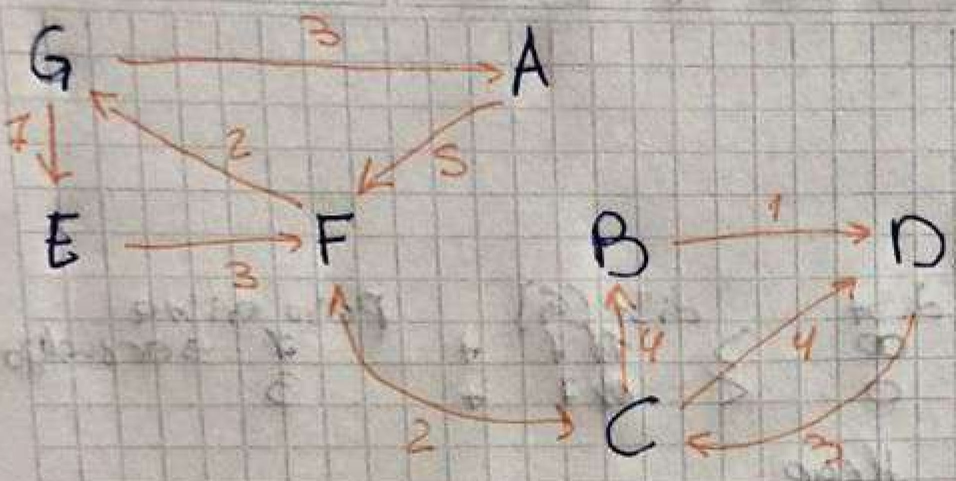
Al ir por cada adyacente: (E veces)

Actualizar los diccionarios → O(1)

Encolar y desencolar del heap (Ya que en el heap se encola como máximo E veces (1 vez por cada adyacente al que se le mejora la distancia)) → O(logE) → O(logV²) → O(2logV) → O(logV)

⇒ O(E logV)

Total ⟹ O(V + E logV)




de minimos

Padres: A : None

Dists
A : 0
B : ∞
C : ∞
D : ∞
E : ∞
F : ∞
G : ∞

Heap: (A 0)

Desencolo → (A 0)

Adyacentes de A → F   dist ∞ > dist[A] 0 + Peso(A,F) 5
↳ Actualizo

Padres
A: None
F: A

Dist
A: 0
F: 5
(el resto igual)

Heap: → (encolo a F)
(F 5)
↳ dist [F]

Desencolo → (F 5)

Adys de F → G   dist ∞ > dist[F] 5 + Peso(F, ady) 2
C   ∞ > 5 + 2
↳ actualizo ambos

Padres
A: 0
F: ~~7~~ A
C: ~~7~~ F
G: ~~7~~ F

Dist
A: 0
F: 5
C: 7
G: 7

Heap: (C 7) (G 7)

Desencolo → (C 7)

Adys de C → F   5 < 7 + 2 ①
B   ∞ > 7 + 4
D   ∞ > 7 + 4
} actualizo

① F ya fue desencolado → ya no se puede mejorar

| Padres | Dist | Heap |
|---|---|---|
| A : None | A 0 | (G 7) (B 11) (D 11) |
| F : A | F 5 | |
| C : F | C 7 | |
| G : F | G 7 | |
| B : C | B 11 | |
| D : G | D 11 | |

Desencolo → (G 7)

| | | dist | | dist [G] | | peso arista | |
|---|---|---|---|---|---|---|---|
| Adys de g → | E | ∞ | > | ~~11~~ 7 | + | 7 | → actualizo |
| | A | 0 | < | ~~11~~ 7 | + | 3 | |

| Padres | Dist | Heap |
|---|---|---|
| A - None | A 0 | (B 11) (D 11) (E ~~18~~ 14) |
| F - A | F 5 | |
| C - F | C 7 | |
| G - F | G 7 | |
| B - C | B 11 | |
| D - C | D 11 | |
| E - G | E ~~18~~ 14 | |

Desencolo → (B 11)

Adys → D 11 < 11 + 1
↳ no hago nada

Desencolo → (D 11)

Adys → C ~~11~~ 7 < 11 + 3
↳ no Hago nada

Desencolo → (E ~~18~~ 14)

Adys → F 5 < 18 + 3
↳ no hago nada

Heap vacio ←

| PADRES | Distancias |
|---|---|
| A: None | A: 0 |
| B: C | B: 11 |
| C: F | C: ~~11~~ 7 |
| D: C | D: 11 |
| E: G | E: 14 |
| F: A | F: 5 |
| G: F | G: 7 |