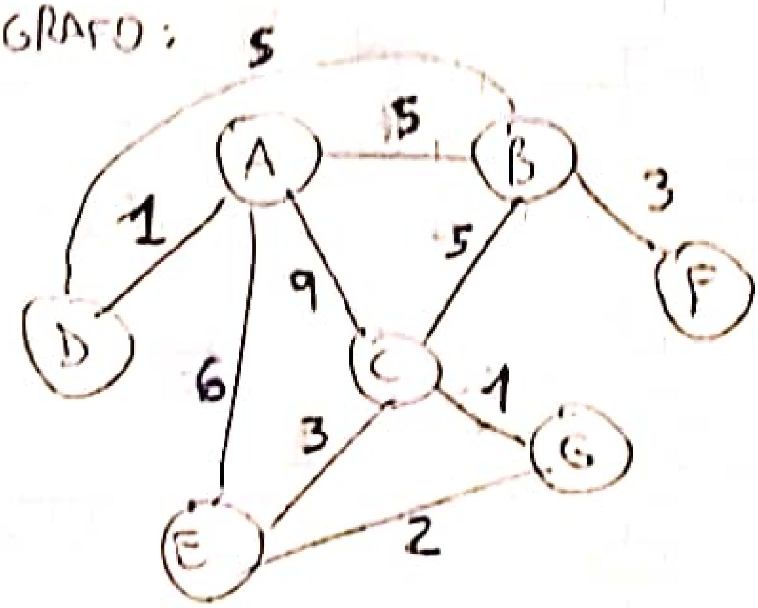
1082444

2) Seguimiento DIJKSTRA

GRAFO:



1) Crear diccionarios de distancia y de padres, y el heap de mínimos

DISTANCIA	PADRES

DISTANCIA = Dis
PADRES = Pad
MÍNIMOS = Min

MÍNIMOS [

2) Elijo un vértice al azar, en este caso, D, y lo inicializo

Dis	Pad
D: 0	D: None

3) Colocamos todos los demas vértices a distancia ∞

Dis	Pad
~~D 0~~	~~D: None~~
A ∞	A
B ∞	B
C ∞	C:
D 0	D: None
E: ∞	E:
F ∞	F:
G ∞	G:

V3

(4) Encolo el primer elemento en el heap (NODO, DIST. AL ORIGEN)

M [(D,0)]

(5) Quedan elementos en el heap? SÍ → desencolo y miro sus adyacentes

ADYACENTES de (D) : (A), (B)

(6) Elegimos un adyacente de (D) por ej. (A), y preguntamos:

$$\text{dist}[A] > \text{dist}[D] + \text{peso}[D,A] ?$$
$$\infty > 0 + 1$$

(7) Como se cumple la ecuación, actualizamos el valor de (A)

dis	pad
A: 1	A: D
D: 0	D: None

(8) Como mejoramos (A), la encolamos:

M [(A,1)]

(9) Ahora miramos (B): dist B > dist D + peso (D,B) ?

$$\infty \quad 0 + 5$$

(10) Cumple ecuación, actualizamos (B):

dis	pad
A: 1	A: D
D: 0	D: None
B: 5	B: D

(11) Encolamos (B) → M [(A,1), (B,5)]

108244

12) Quedan elementos en el heap ? Sí → desencolamos y miramos sus ady.

$\text{H}\,[\cancel{(A,1)}, (B,5)$

↘ $(A, 1)$ → adyacentes de A : B, C, E

13) para B : dist B ≥ dist A + peso (A,B) ?

$5 > 1 + 5$ ✗

14) No hacemos nada con B

15) para C : dist C > dist A + peso (A, C) ?

$\infty > 1 + 9$ ✓

16) Actualizamos C

dis	pad
A: 1	A: D
B: 5	B: D
C: 10	C: A
D: 0	D: None

17) Encolamos C

$\text{H}\,[(B,5), (C,10)]$

18) Para E : dist E > dist A + peso (A,E) ?

$\infty > 1 + 6$

~~19) Actualizamos E~~

Escaneado con CamScanner

/3

19) Actualizamos E:

dis	pad
A: 1	A: D
B: 5	B: D
C: 10	C: A
D: 0	D: None
E: 7	E: A

20) Encolamos E: M[(B,5), (C,10), (E,7)]

21) Terminamos con A, desencolamos el heap:

M[~~(B,5)~~, (C,10), (E,7)] → (B,5) → adyacentes B: A

22) A: dist A > dist B + peso B, A ? ✗

23) C: dist C > dist B + peso B, C ?

10 > 5 + 5 ✗

24) F: dist F > dist B + peso B, F ? ✓

∞ > 5 + 3

25) Actualizamos F:

dis	pad
A: 1	A: D
B: 5	B: D
C: 10	C: A
D: 0	D: None
F: 8	F: B
E: 7	E: A

26) Encolamos F → M[(C,10), (E,7), (F,8)]

27) Terminamos con B, desencolamos el heap

M[(C,10), ~~(E,7)~~, (F,8)]

→ (E,7) → adyacentes E: A

108244

28) A: (omitimos)

C: dist C > dist E + peso E,C ? ✗

$10 > 7 + 3$

G: dist G > dist E + peso E,G ? ✓

$\infty > 7 + 2$

29) Actualizamos G:

dis	pad
A: 1	A: B
B: 5	B: D
C: 10	C: A
D: 0	D: None
E: 7	E: A
F: 8	F: B
G: 9	G: E

30) Encolamos G: M[(C,10), (F,8), (G,9)]

31) Terminamos con E, desencolamos el heap:

M[(C,10), $\cancel{(F,8)}$, (G,9)]

↳ (F,8) → ady. F: B

32) Para B: (omitir, visitado)

33) Terminamos con F, desencolamos el heap

M[(C,10), $\cancel{(G,9)}$]

↳ (G,9) → ady. G: E, C

34) para (E). (omitir, visitado)

35) para (E): dist C > dist G + peso G, C ?
10 > 9 + 1 X

36) Terminamos con (G), desencolamos el heap
M [~~(C,10)~~] → (C,10) → ady. (C): A, B, E, G

37) (Omitimos A, B, E, G porque ya fueron visitados)

38) Terminamos con (C). El heap tiene elementos? NO

39) Terminamos el algoritmo.

RESULTADOS FINALES:

DISTANCIA	PADRE
A: 1	A: D
B: 5	B: D
C: 10	C: A
D: 0	D: None
E: 7	E: A
F: 8	F: B
G: 9	G: E