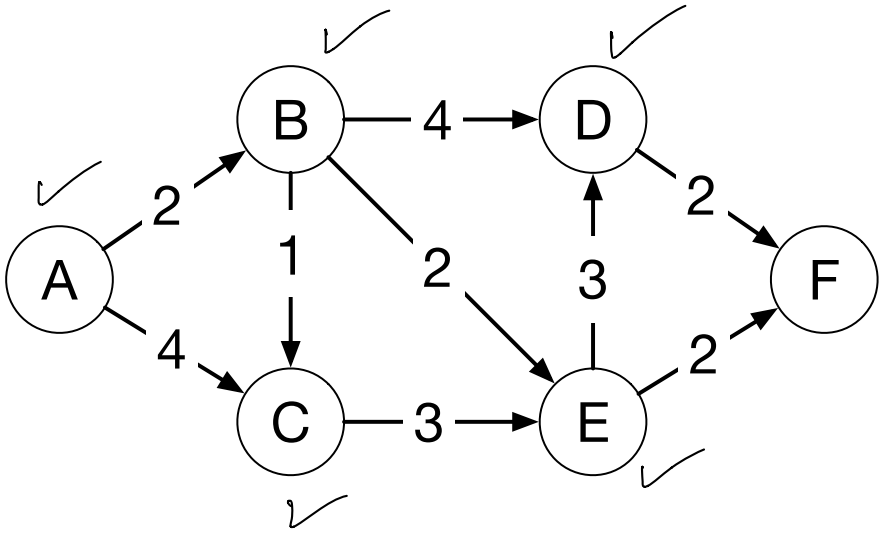
Find the shortest path from A to F.



Priority queue

~~(A, 0)~~

~~(B, 2)~~

~~(C, 4)~~ ~~(C, 3)~~

~~(D, 6)~~

~~(E, 4)~~

~~(F, 6)~~

Order that we visit vertices: A, B, C, E, D, F

Dist table

dist[A]: ~~∞~~ 0

dist[B]: ~~∞~~ 2

dist[C]: ~~∞~~ ~~4~~ 3

dist[D]: ~~∞~~ 6

dist[E]: ∞ 4

dist[F]: ~~∞~~ 6

Prev table

prev[A]: X

prev[B]: ~~X~~ A

prev[C]: X ~~A~~ B

prev[D]: X B

prev[E]: X B

prev[F]: X E

u → v: A → B

dist[u] ? ∞

weight(u,v) = 2

alt = 0 + 2

dist[v] ? ∞

u → v: A → C

dist[u] 0

weight = 4

0 + 4 < ∞

Final shortest path distance: 6

Final shortest path: F - E - B - A → Reverse → A B E F

u → v: B → C

dist[u] = 2

weight = 1

2 + 1 < 4

u → v: E → F

4 + 2 = 6

u → v: C → E

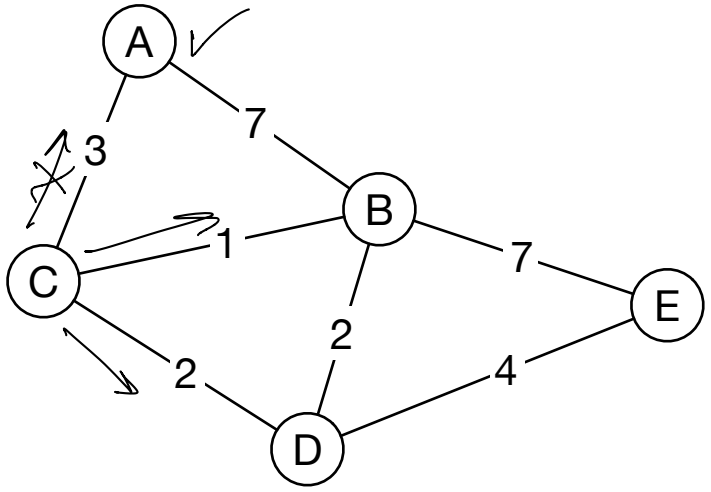
3 + 3 < 4

u → v: D → F

6 + 2 < 6 X

u → v: e → d

4 + 3 < 6 X



Find the shortest path from A to E.

Priority queue

~~(A, 0)~~

~~(C, 3)~~

(B, 7)

Order that we visit vertices: A, C, B, D, E

C → B
u v

dist[u] + weight (u,v) < dist[v]

3 + 1 < 7

Dist table		Prev table	
dist[A]:	~~0~~	prev[A]:	X
dist[B]:	~~7~~ 4	prev[B]:	~~X~~ A C
dist[C]:	~~3~~	prev[C]:	X
dist[D]:	∞	prev[D]:	X
dist[E]:	∞	prev[E]:	X

Final shortest path distance: 9

Final shortest path: A - C - D - E