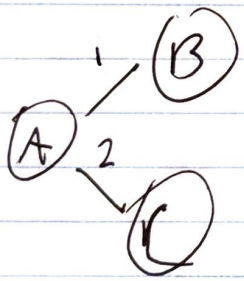
A	Start: A	pop(A)	mark(A)		
BC	push(B)	Mark(B),	push(C)	mark(C)	
CDE	pop(B)	push(D)	push(E)	Mark(D)	mark(E)
DE	pop(C)	~~push(E)~~	~~mark(E)~~		
EF	pop(D)	push(F)	mark(F)		
F	pop(E)				

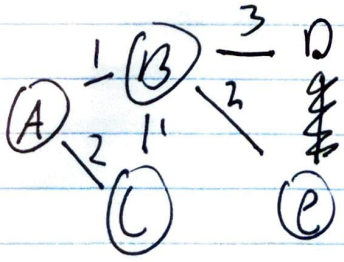
(F) = first in queue

There exists a path



{A, ∞, ∞, ∞, ∞, ∞}

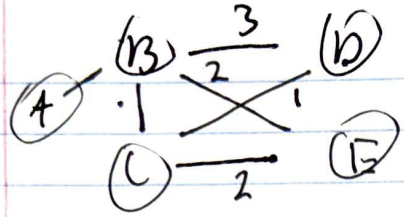
{A} path



{0, 1, 2, ∞, ∞, ∞}

{A, B, C}

we saw b had distance of 1, & c had distance of 2, so spt → B had to be directly to it. Next we see from b → c = 1, a → b → c = 2, = A → c. so we use A → B → c



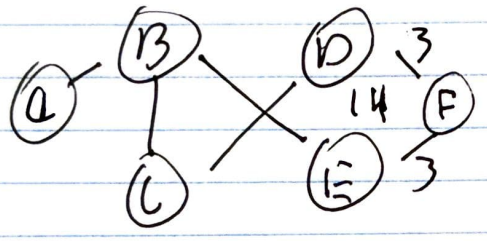
B->D B->E

{4, 3, ∞, ∞, ∞}

{A, B, C}

now we compare B->D B->E with
C->D, C->E
=3 =4

∴ B->C->D & B->E



we then verify D<->E is not
D->E E->D D->F only faster

{8, 6, 8}

{A, B, C, D, E}

Then we choose D, although either work since both are 6