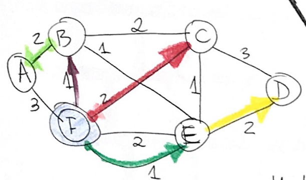

- Y(X, p, q): this is an edge that goes from X to Y. local cost p; global achieved cost q.
- Root node is F

- Overall operation of the algorithm

← start here.

1. Visit node (start at root)
2. Expand visited node:
   - Add edges to bag of temporary edges
3. Select one of the edges in the temporary bag having the least total length
   - Move selected edge to the permanent list and delete it from {T}.
   - Delete all temporary edges that have been already added to {P}

{P} {P} linked list of selected edges | {T} bag of expanded edges {T}

① Visit F from itself at start

F(F, Ø, Ø) - - → ② This is the expansion of F: 5 edges: ~~A(F, 3, 3+Ø)~~ ~~B(F, 1, 1+Ø)~~ ~~C(F, 2, 2+Ø)~~ ~~E(F, 2, 2+Ø)~~ ~~E(F, 1, 1+Ø)~~

③ Select one of the least total cost edges: B(F, 1, 1); E(F, 1, 1): We select E(F, 1, 1)

- select E(F, 1, 1)
- delete E(F, 1, 1) from {T}

① visit E from F

E(F, 1, 1) - - → ② Expand E(F, 1, 1): ~~F(E, 2, 2+1)~~ ~~B(E, 1, 1+1)~~ ~~C(E, 1, 1+1)~~ ~~D(E, 2, 2+1)~~

③ Selection results in B(F, 1, 1) since its global cost is the least in {T}.

- We move B(F, 1, 1) to {P} and delete it (see line in orange above)
- We also delete from {T} any edge that belongs to {P}

B(F, 1, 1) - - → ~~A(B, 2, 2+1)~~ ~~E(B, 1, 1+1)~~ ~~C(B, 2, 2+1)~~

C(F, 2, 2) - - → ~~B(C, 2, 4)~~ ~~E(C, 1, 3)~~ ~~D(C, 3, 5)~~

D(E, 2, 3) → ~~C(D, 3, 6)~~

A(B, 2, 3) - - → ~~F(A, 3, 6)~~ . Now {T} is empty AND algorithm finishes!

Shortest Path tree: