# Algorithms for computing Maximally Redundant Trees for IP/LDP Fast-Reroute
# draft-enyedi-rtgwg-mrt-frr-algorithm-01

Alia Atlas (Juniper Networks)
Gabor Enyedi, Andras Csaszar (Ericsson)

IETF 83, Paris, France

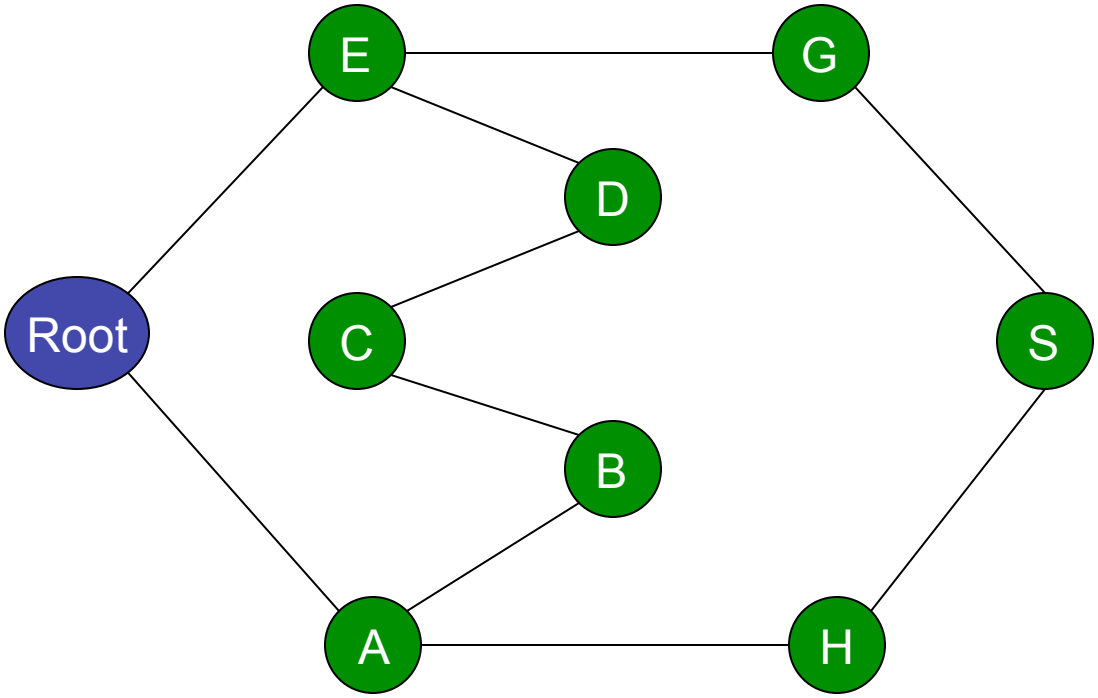
# Agenda

- Briefly about the algorithm
- Problem
- Avoid using a node
- Non-2-connected networks

# Agenda

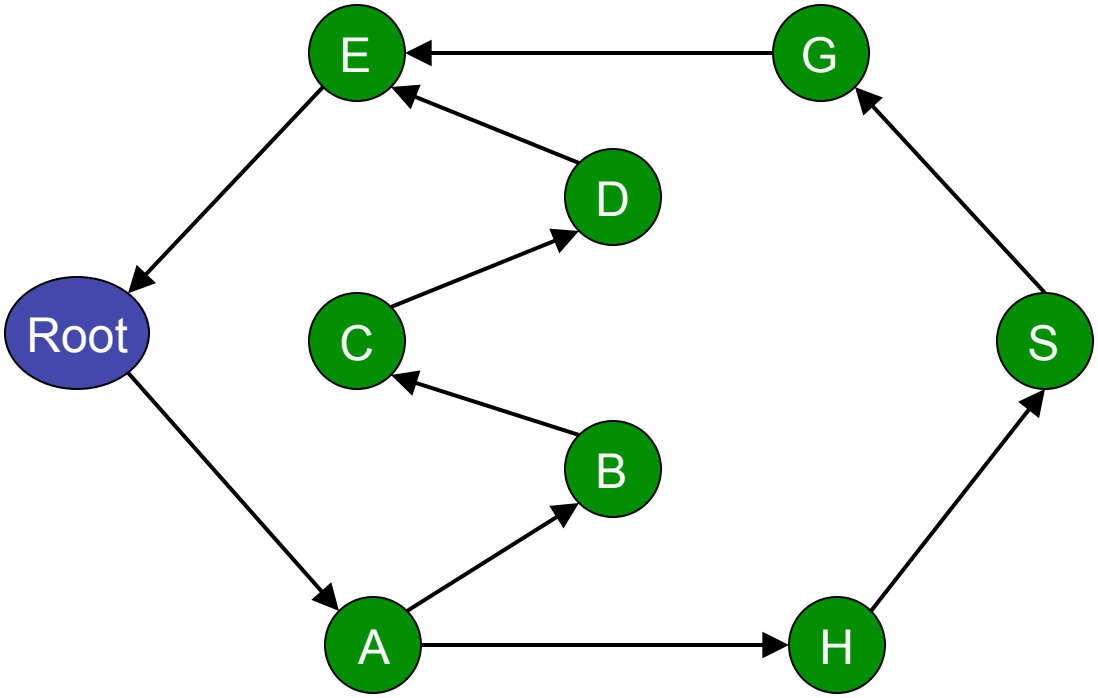
- Briefly about the algorithm
- Problem
- Avoid using a node
- Non-2-connected networks

# ADAG and partial order

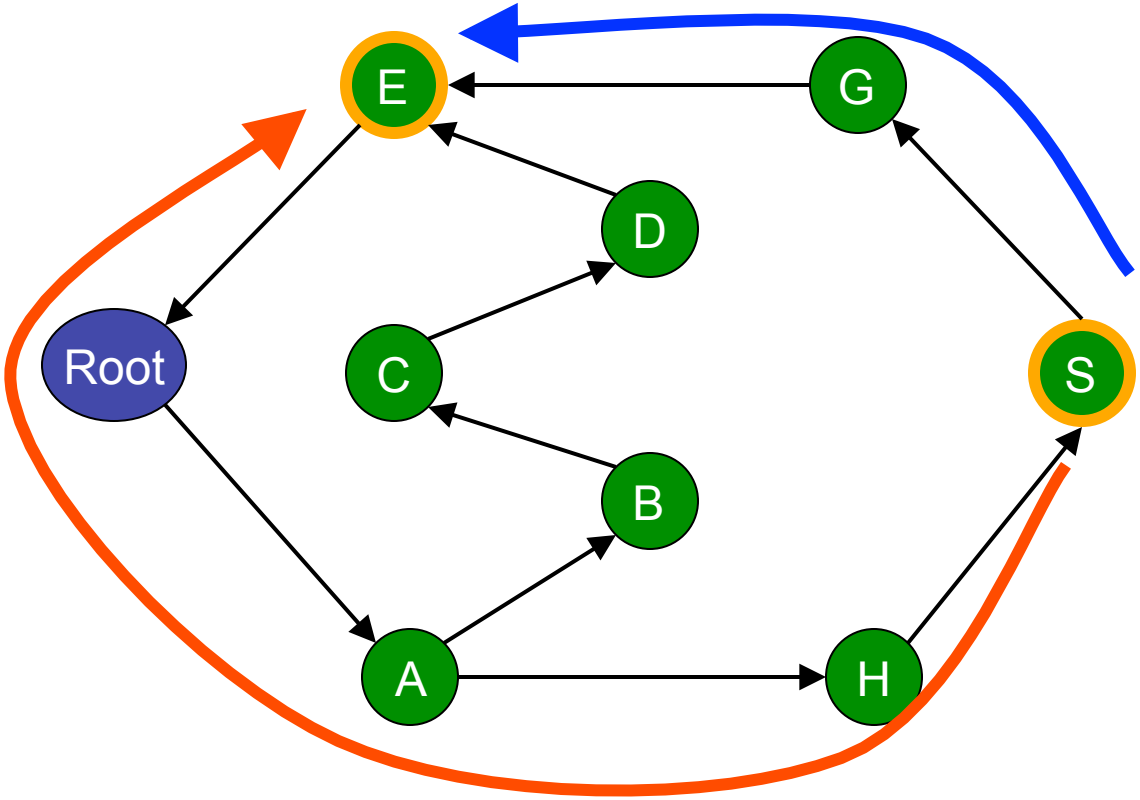


# ADAG and partial order

- Almost DAG (ADAG)
- A<<B if there is a path from A to B
- Root is both the shortest and the greatest

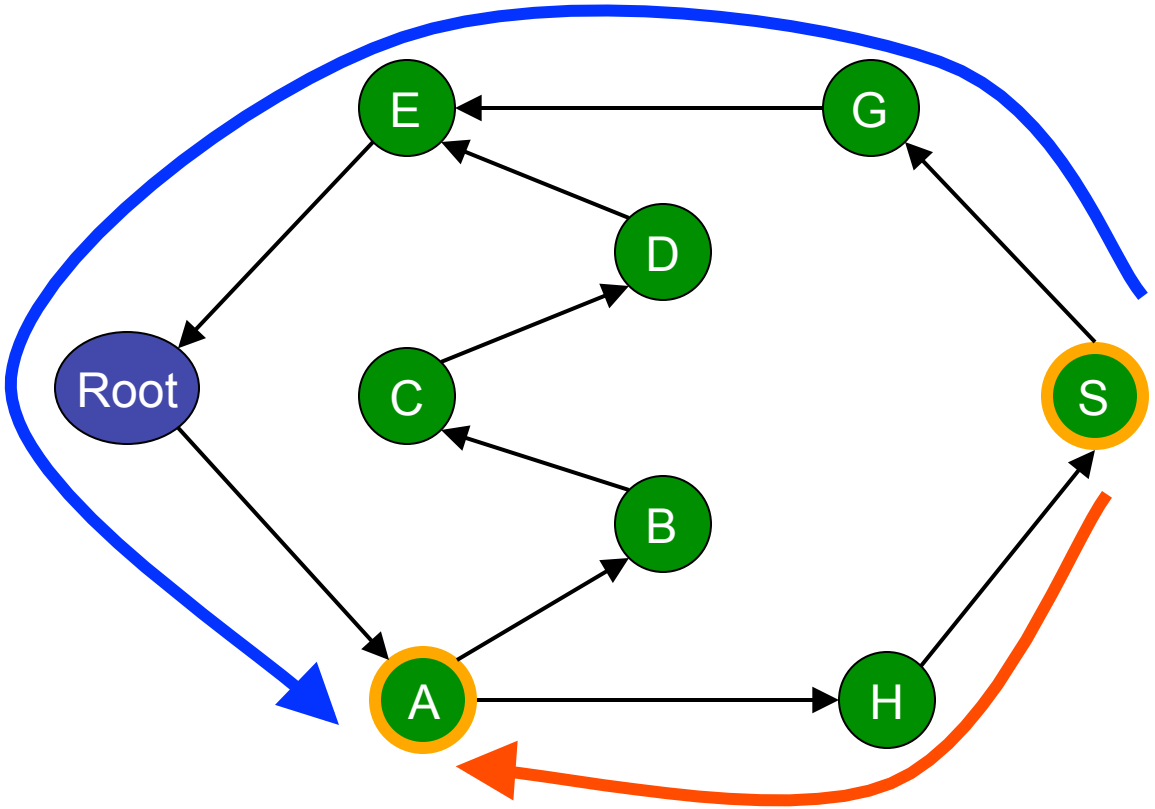


- S<<E
  - Blue path: increasing [S, E]
  - Red path: decreasing [Root, S] and [E, Root]



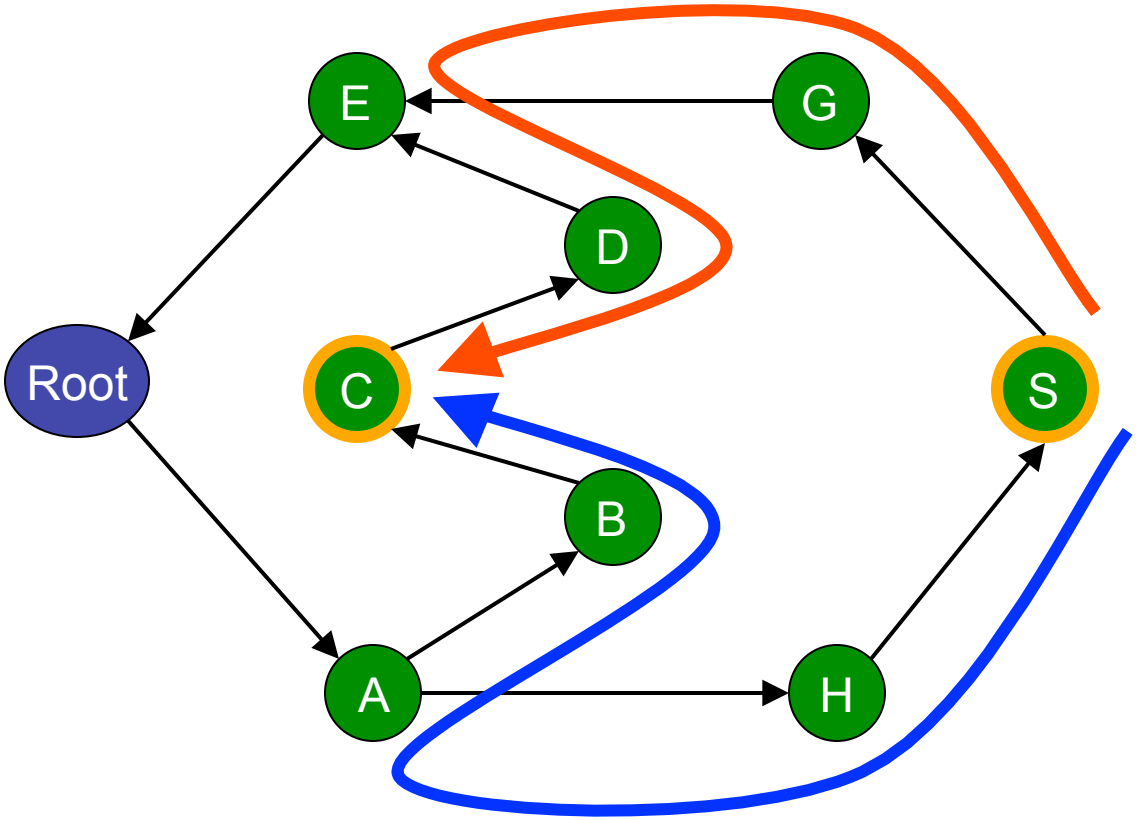
# ADAG and partial order

- S>>A
  - Blue path: increasing [S,Root] and [Root, A]
  - Red path: decreasing [A, S]



# ADAG and partial order

- S and C are not ordered
  - Blue path: [S, E] and [C, E]
  - Red path: [A, S] and [A, C]

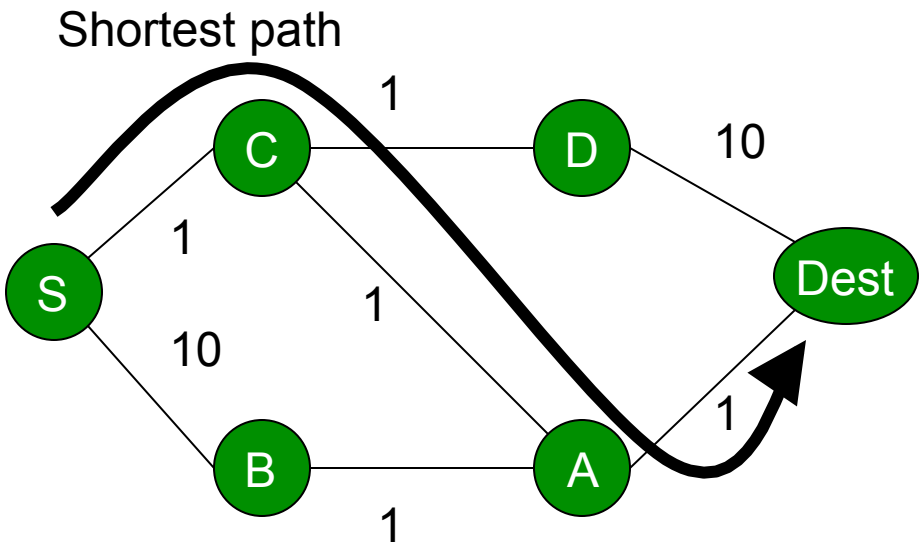


# Agenda

- Briefly about the algorithm
- Problem
- Avoid using a node
- Non-2-connected networks

# Three trees

- We have tree trees
  - SPT
  - Two MRTs
- There is no connection between SPT and MRTs
- Impossible to find a redundant pair for SPT
- Example:




No redundant pair for that!

# Agenda

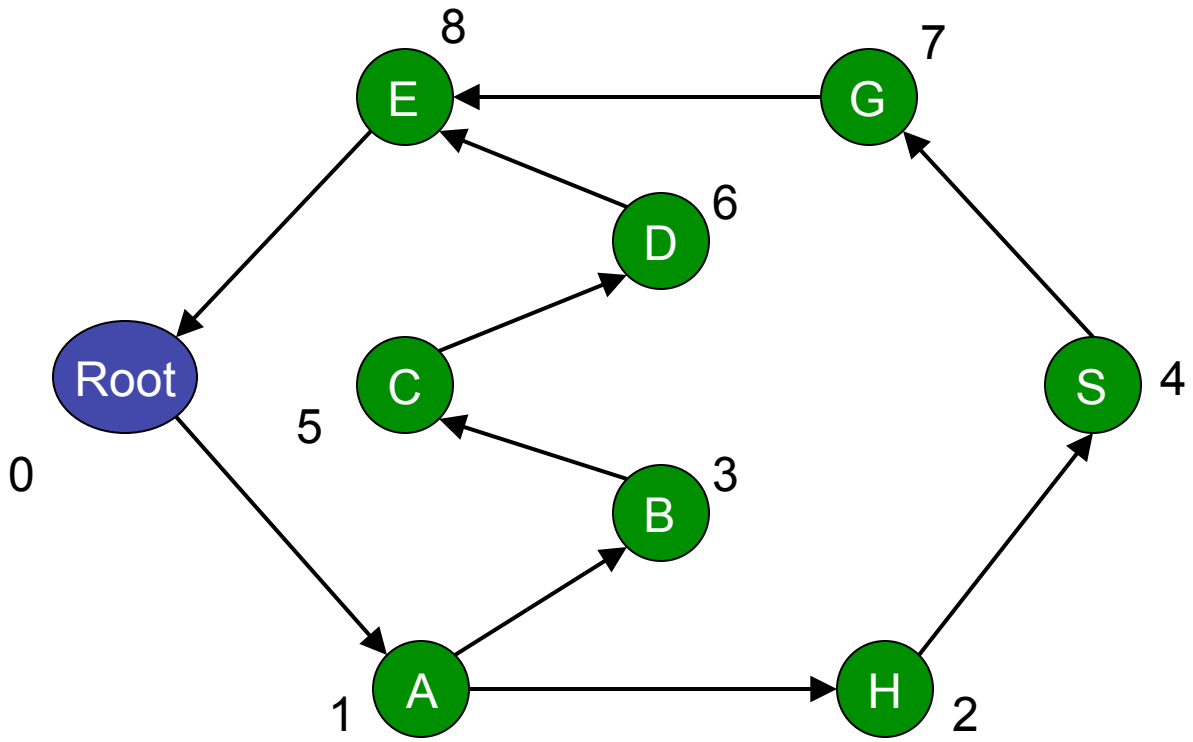
- Briefly about the algorithm
- Problem
- Avoid using a node
- Non-2-connected networks

# Total order

- Partial order can compare any X only with S
  - We need to compare any two nodes
- Make a total order as well
  - If A<<B, let A<B
  - If A and B are not ordered select either A<B or B<A
  - This can be done with a topological oder after converting the ADAG into a DAG
- Results:
    - If A<B, either A<<B or A and B are not ordered

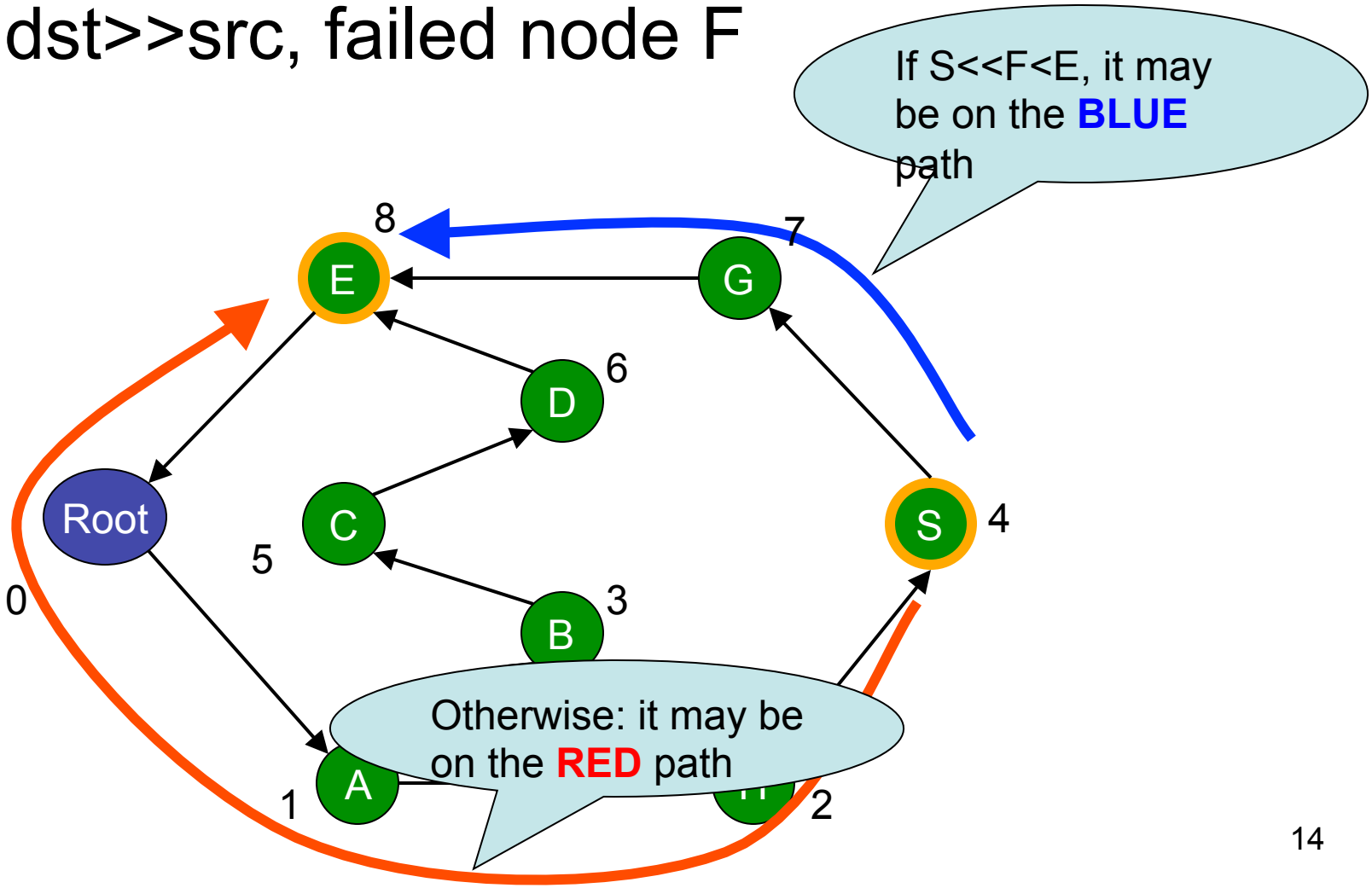
# A possible total order

- Numbers are written next to nodes



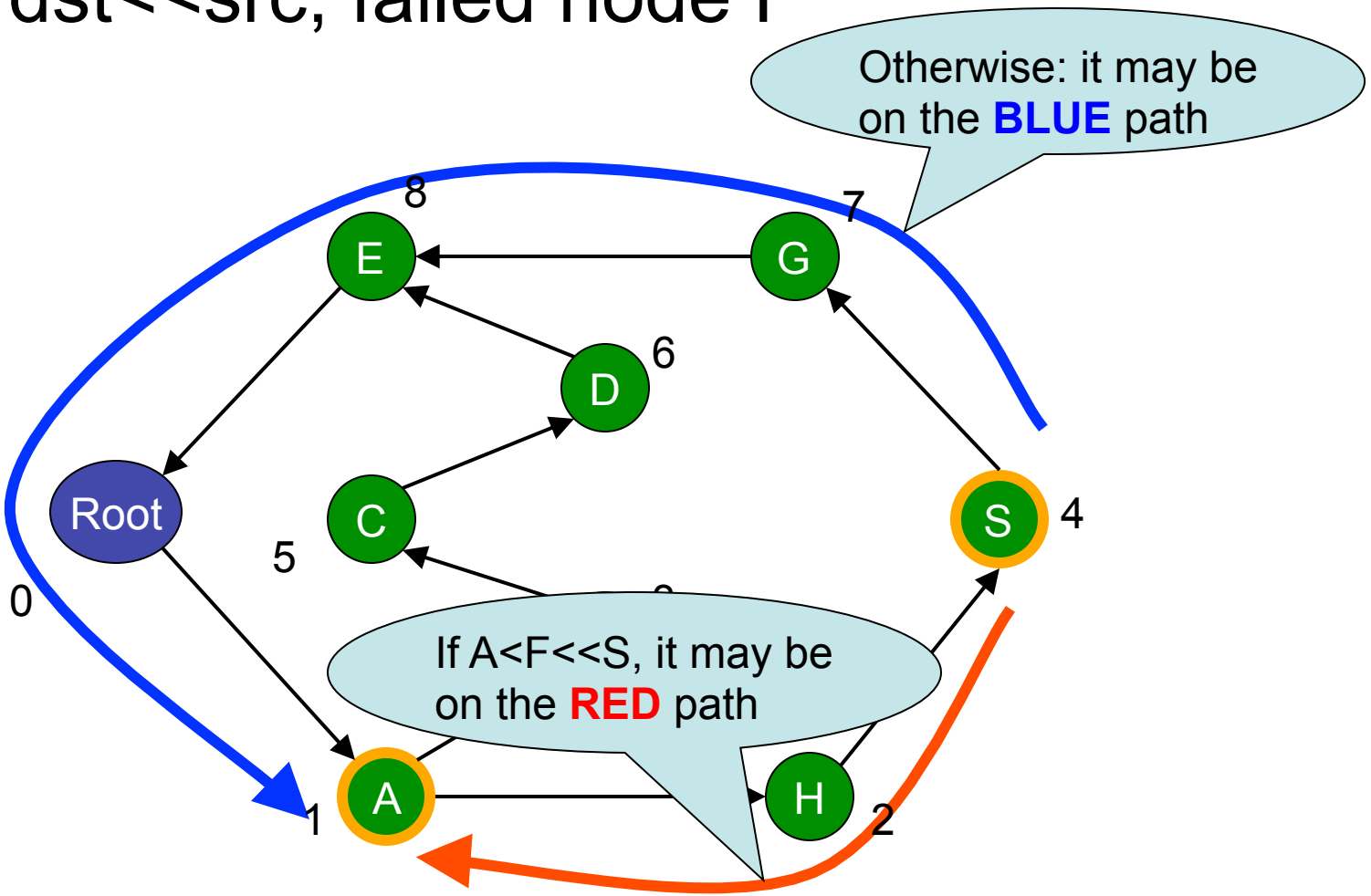
# Possible cases

- If dst>>src, failed node F



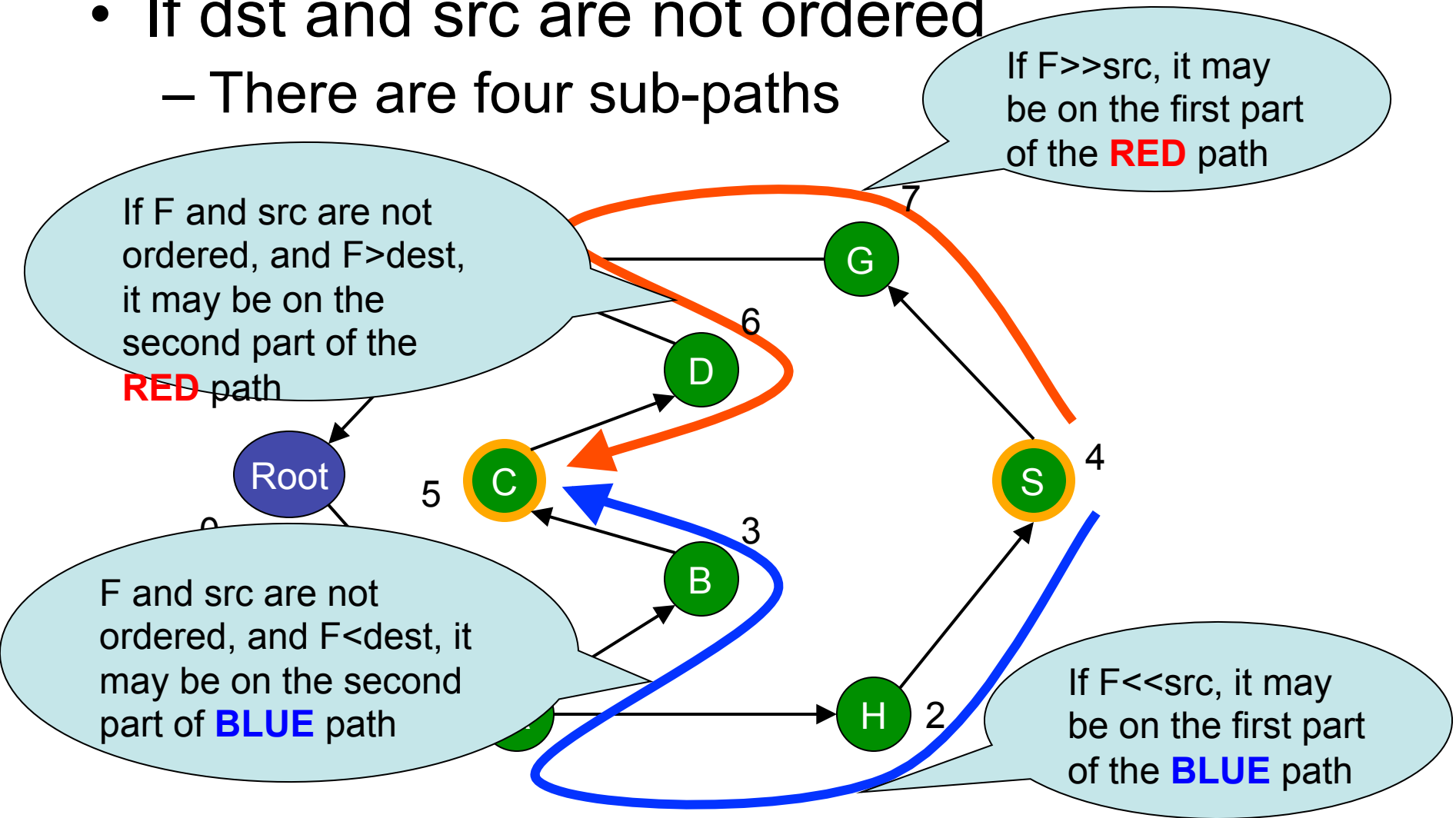
# Possible cases

- If dst<<src, failed node F



# Possible cases

- If dst and src are not ordered
  - There are four sub-paths

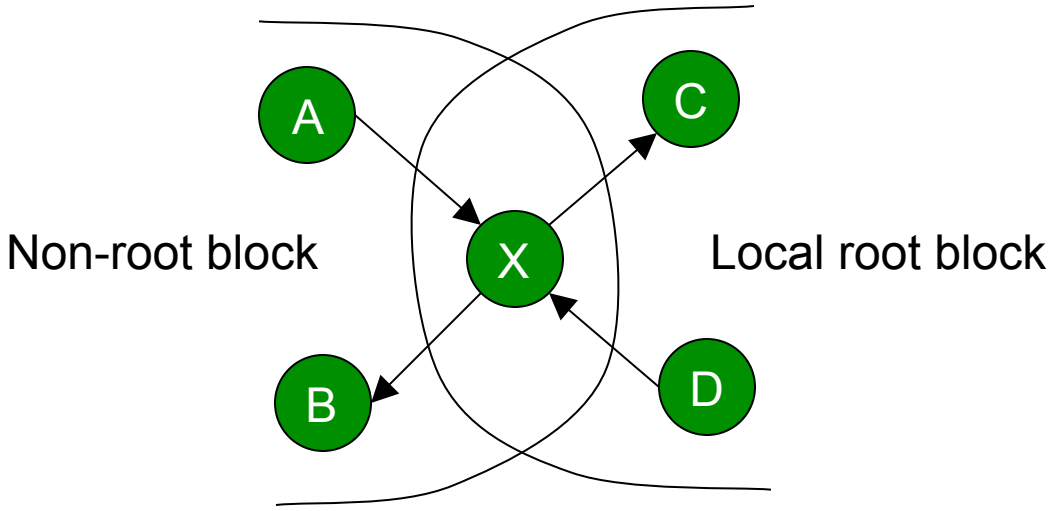


# Agenda

- Briefly about the algorithm
- Problem
- Avoid using a node
- Non-2-connected networks

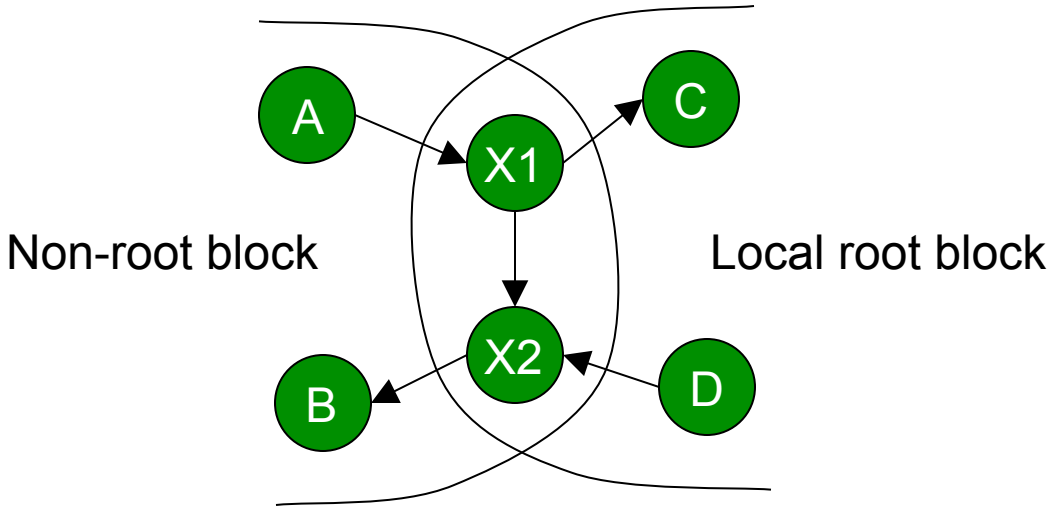
# Non-2-connected problem

- In this case we don't have a single order
  - Neither a partial order
  - Nor a total order
- Convert the GADAG into an ADAG!



# Non-2-connected problem

- In this case we don't have a single order
  - Neither a partial order
  - Nor a total order
- Convert the GADAG into an ADAG!



*Thank you!*