Exercises: Artificial Intelligence

A*

A*

A* ALGORITHM

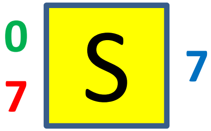
A* Algorithm

- ***Input:***
  - **QUEUE**: Path only containing root
- ***Algorithm:***
  - **WHILE** (QUEUE not empty && first path not reach goal) **DO**
    - Remove ***first path*** from QUEUE
    - Create paths to all children
    - Reject paths with loops
    - Add paths and sort QUEUE (by f = cost + heuristic)
    - **IF** QUEUE contains paths: P, Q
      AND P ends in node N_i && Q contains node N_i
      AND cost P $\geq$ cost Q
      THEN remove P
  - **IF** goal reached **THEN** success **ELSE** failure

A*

FIRST EXAMPLE ON A*

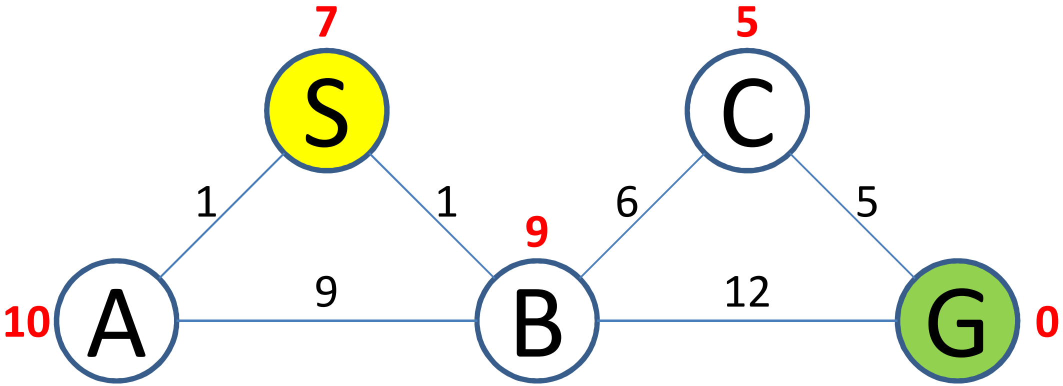
A* algorithm by Example



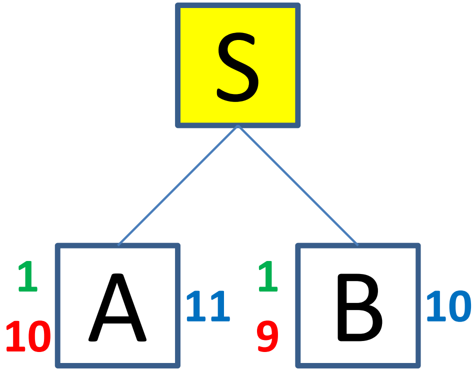
f = accumulated path cost + heuristic

QUEUE = ***path containing root***

QUEUE: <S>



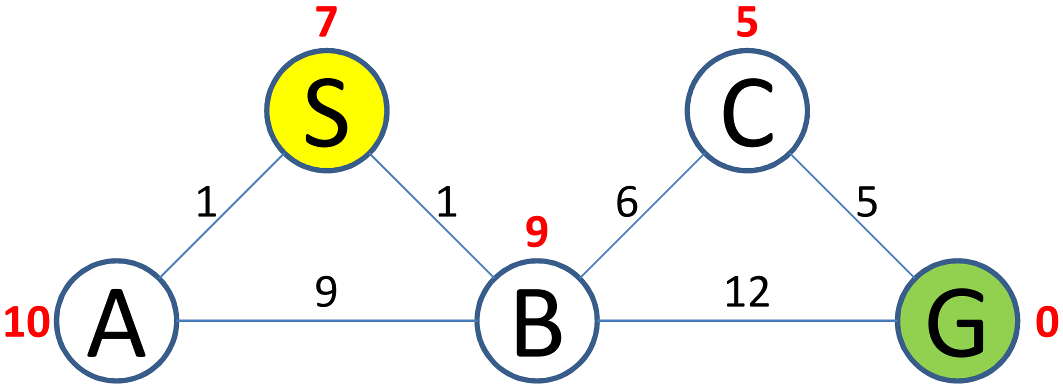
A* algorithm by Example



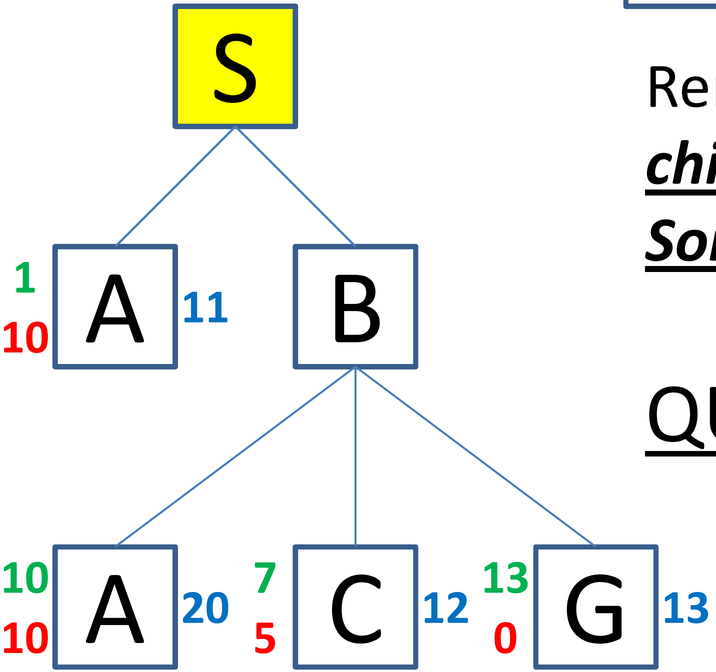
f = accumulated path cost + heuristic

Remove ***first path***, Create ***paths to all children***, Reject ***loops*** and ***Add paths***. ***Sort*** QUEUE ***by f***

QUEUE: <SB,SA>



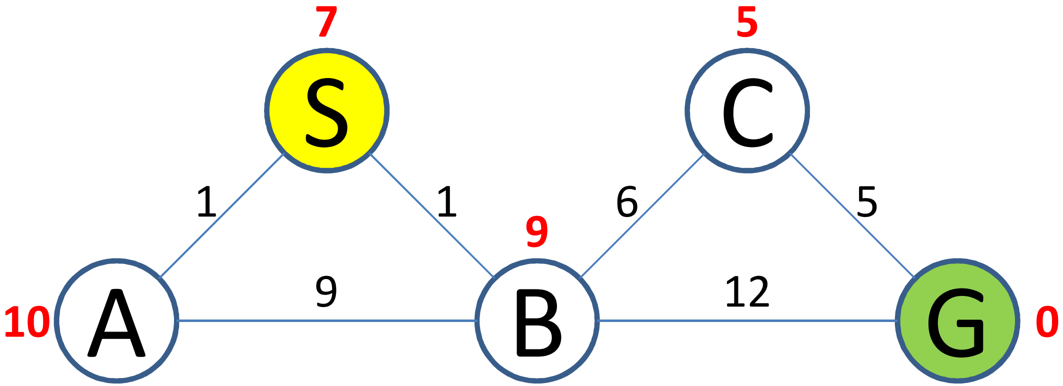
A* algorithm by Example



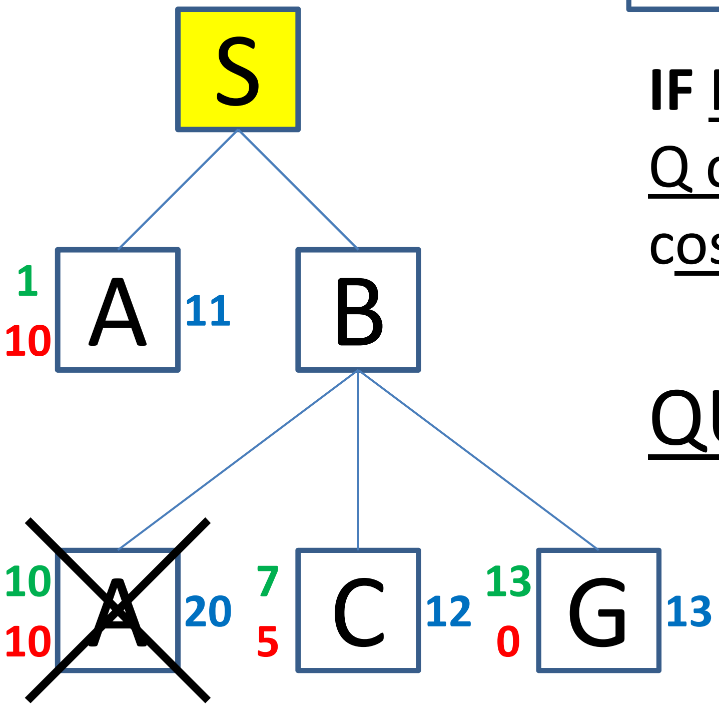
f = accumulated path cost + heuristic

Remove ***first path***, Create ***paths to all children***, Reject ***loops*** and ***Add paths***. ***Sort*** QUEUE ***by f***

QUEUE: <SA,SBC,SBG,SBA>



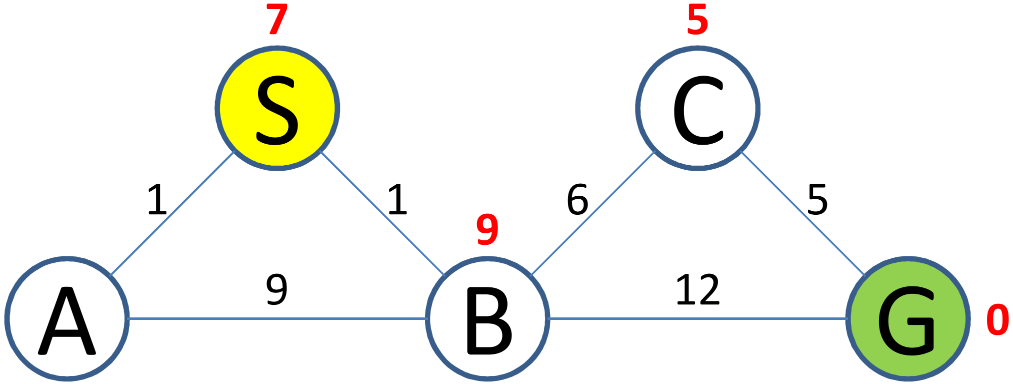
A* algorithm by Example



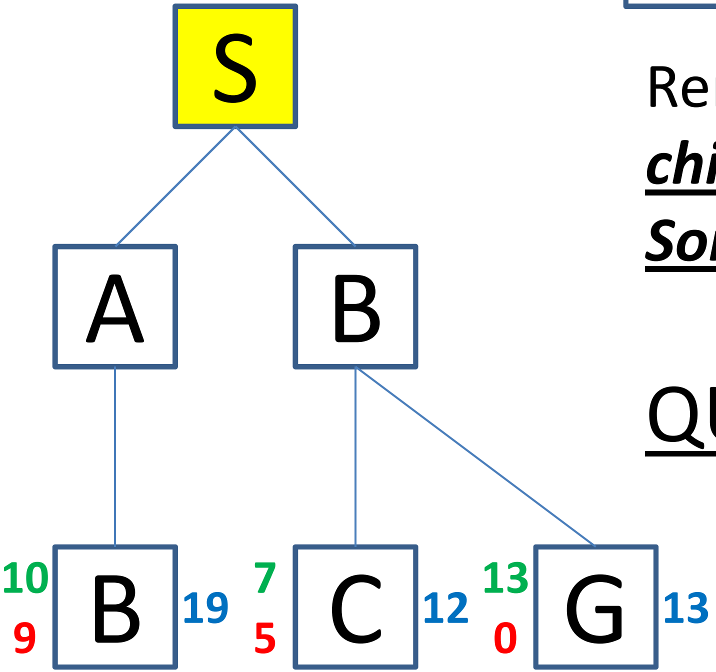
f = accumulated path cost + heuristic

IF P terminating in I with cost_P && Q containing I with cost_Q **AND** cost_P ≥ cost_Q **THEN** remove P

QUEUE: <SA,SBC,SBG,**SBA**>



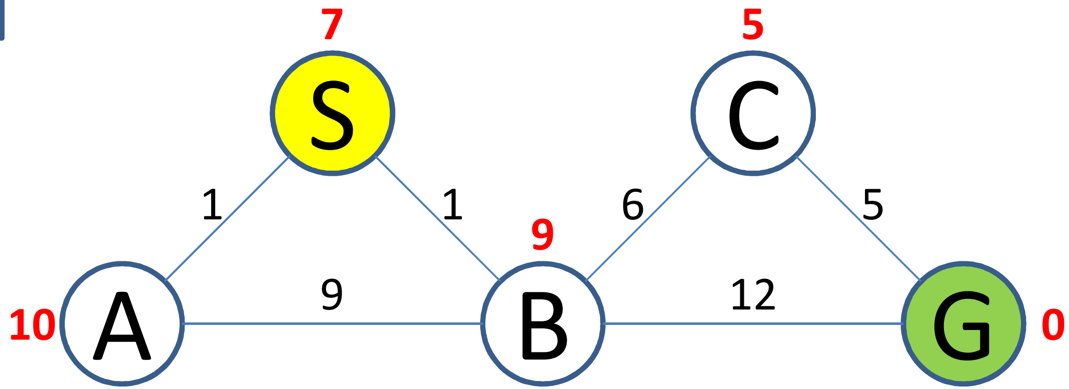
A* algorithm by Example



f = accumulated path cost + heuristic

Remove ***first path***, Create ***paths to all children***, Reject ***loops*** and ***Add paths***. ***Sort*** QUEUE ***by f***

QUEUE: <SBC,SBG,SAB>

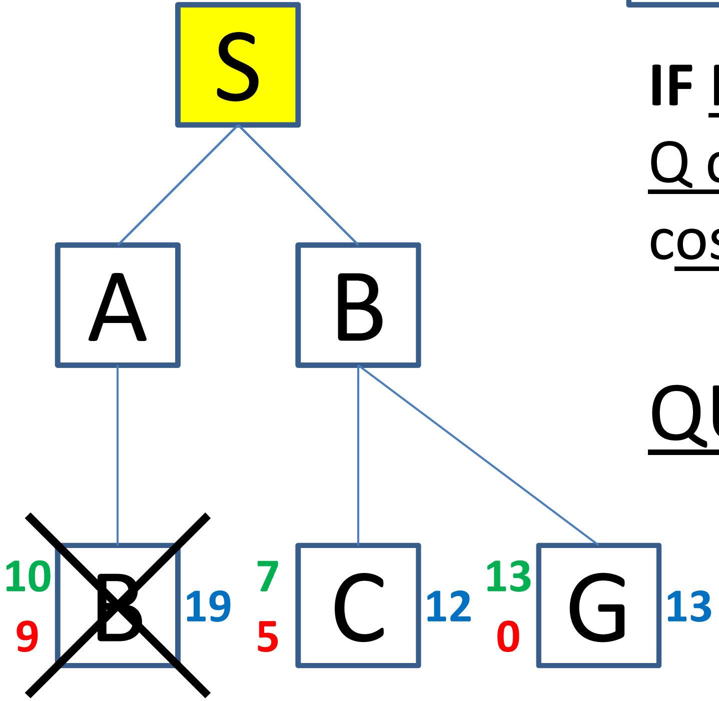


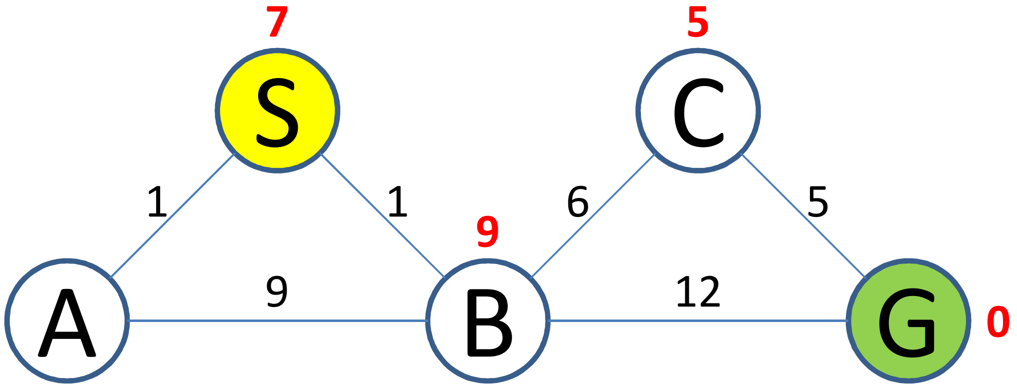
A* algorithm by Example

f = accumulated path cost + heuristic

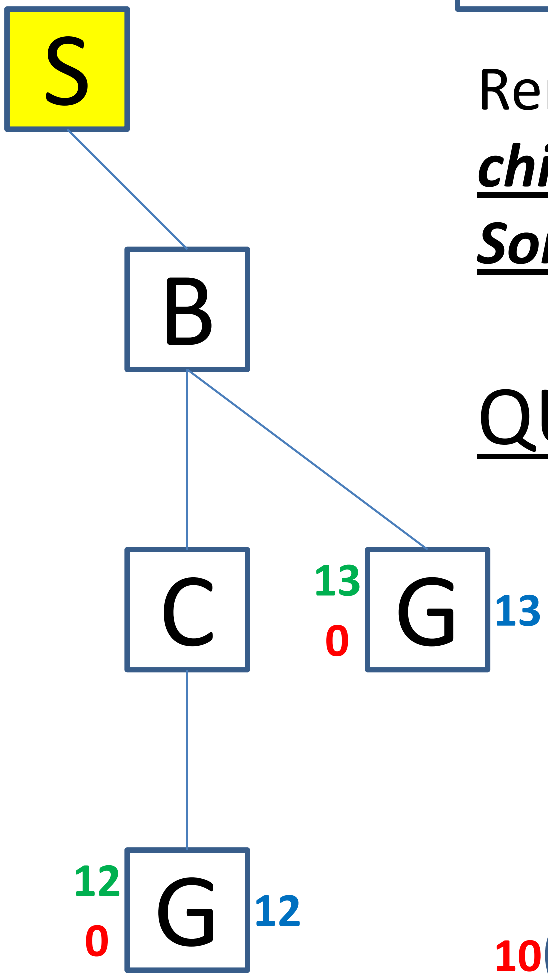
IF P terminating in I with cost_P && Q containing I with cost_Q **AND** cost_P ≥ cost_Q **THEN** remove P

QUEUE: <SBC,SBG,**SAB**>





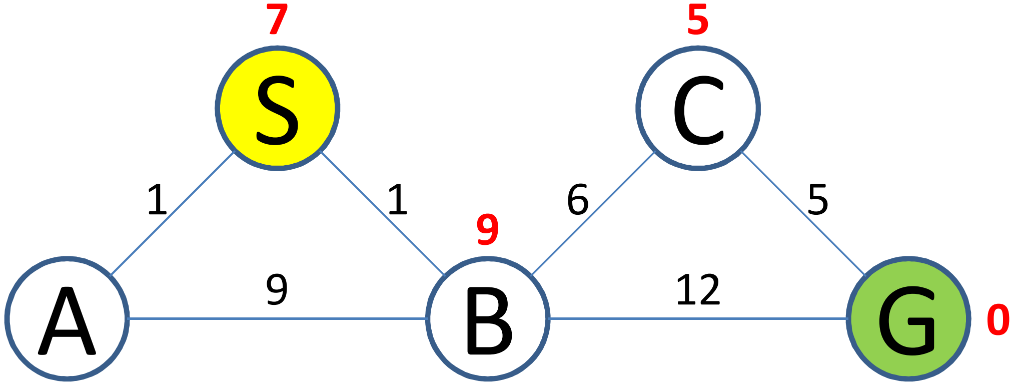
A* algorithm by Example



f = accumulated path cost + heuristic

Remove ***first path***, Create ***paths to all children***, Reject ***loops*** and ***Add paths***. ***Sort*** QUEUE ***by f***

QUEUE: <SBCG,SBG>

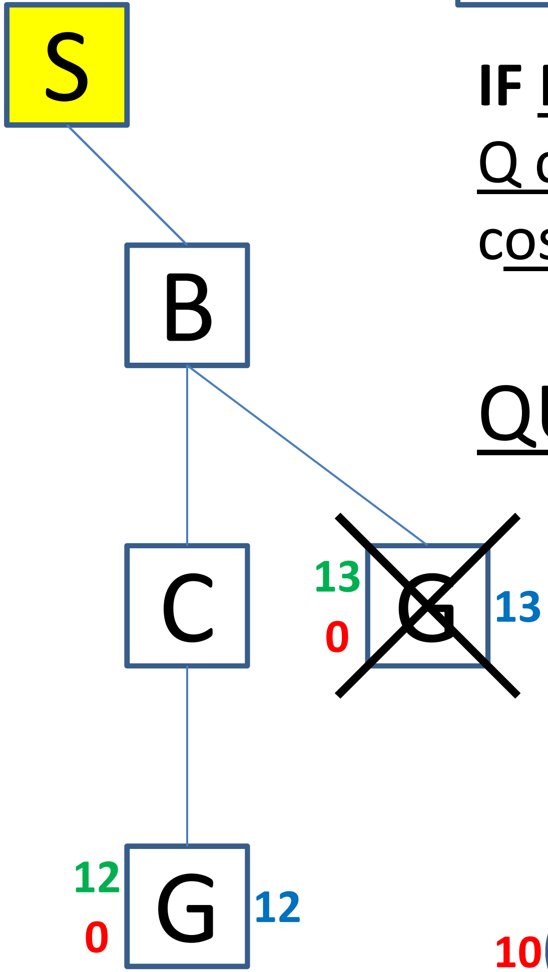


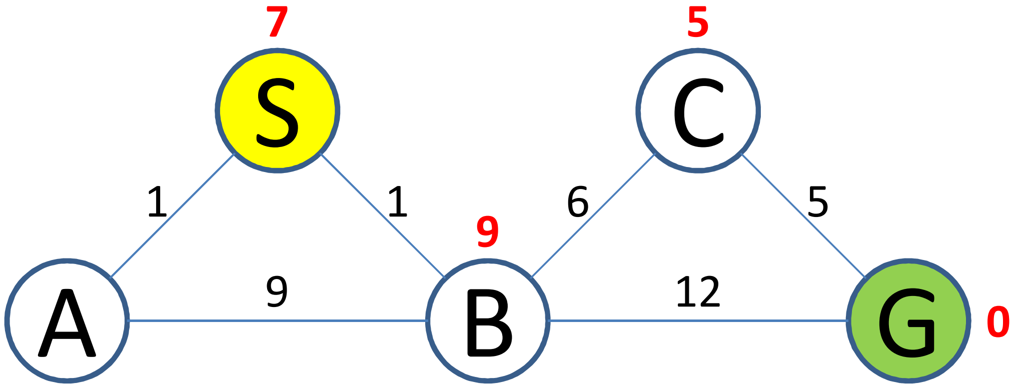
A* algorithm by Example

f = accumulated path cost + heuristic

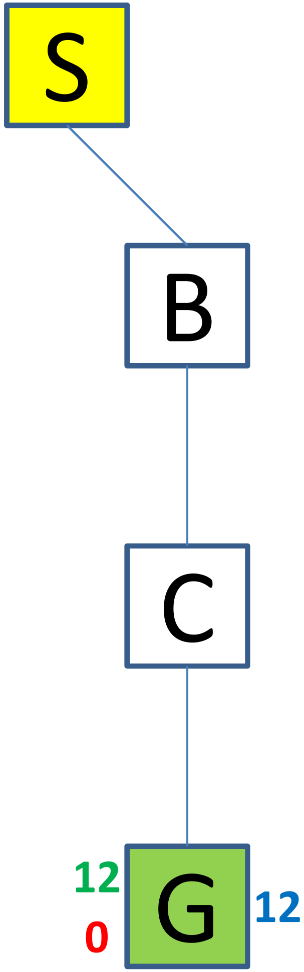
IF P terminating in I with cost_P && Q containing I with cost_Q **AND** cost_P ≥ cost_Q **THEN** remove P

QUEUE: <SBCG,**SBG**>





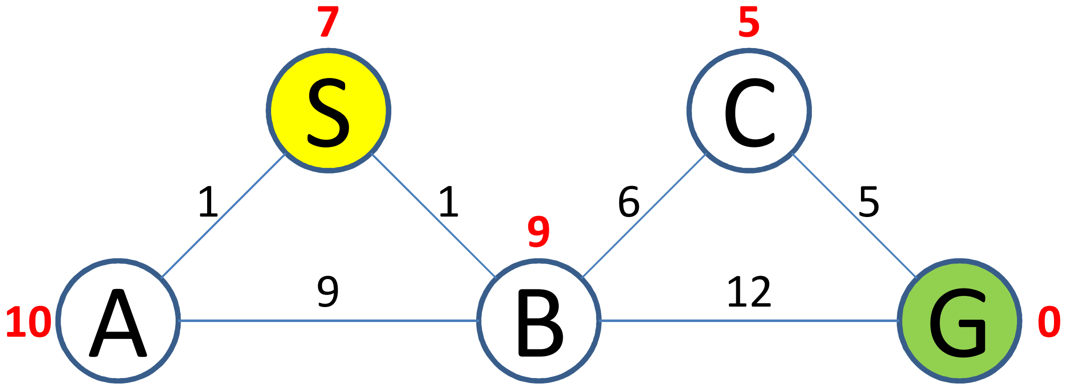
A* algorithm by Example



f = accumulated path cost + heuristic

SUCCESS

QUEUE: <SBCG>

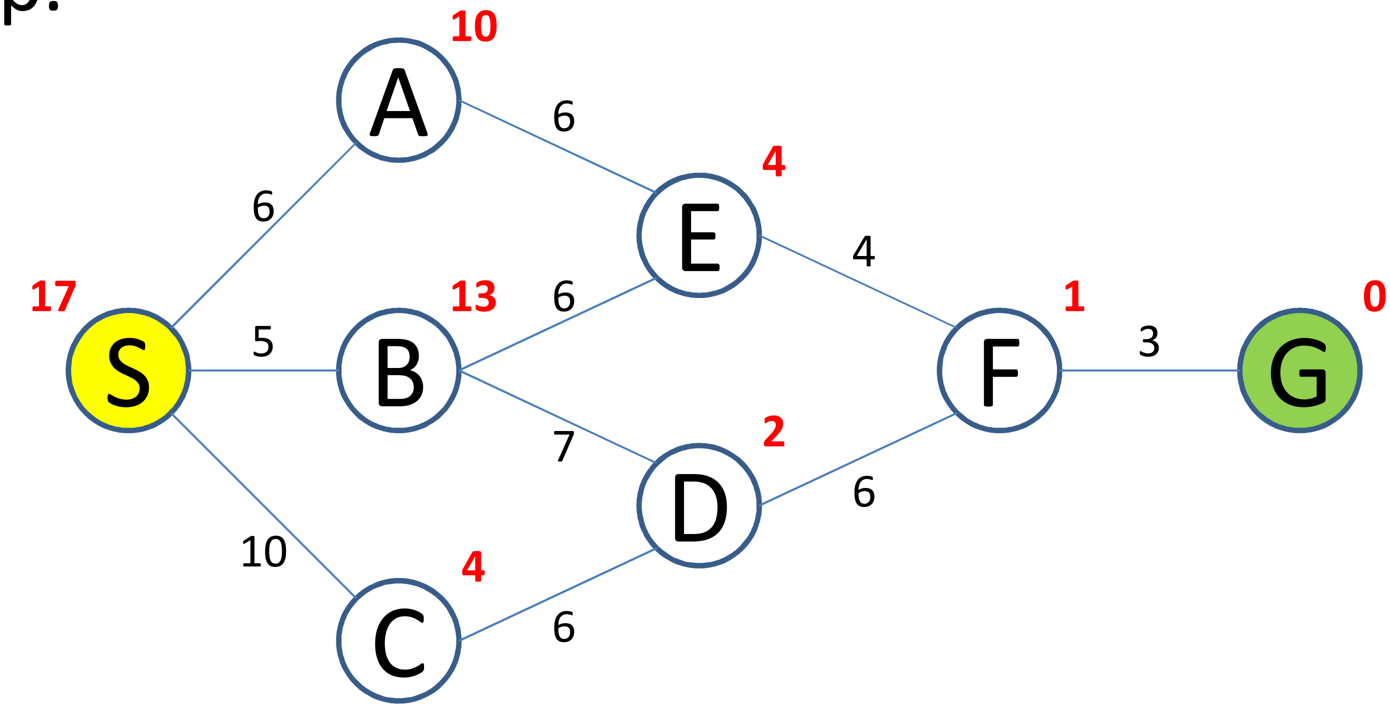


A*

PROBLEM

Problem

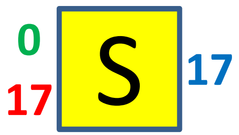
- Perform the A* Algorithm on the following figure. Explicitly write down the queue at each step.

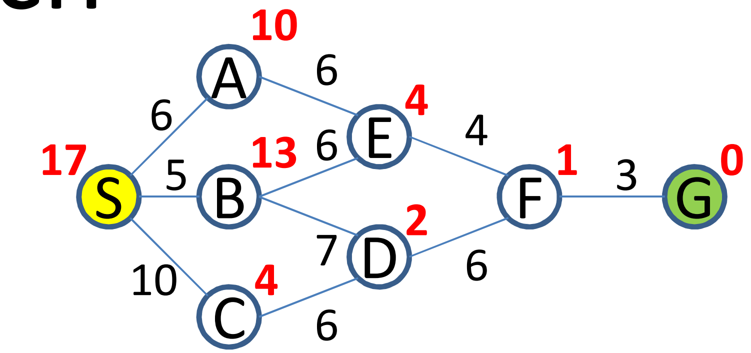


A*

A* SEARCH

A* Search

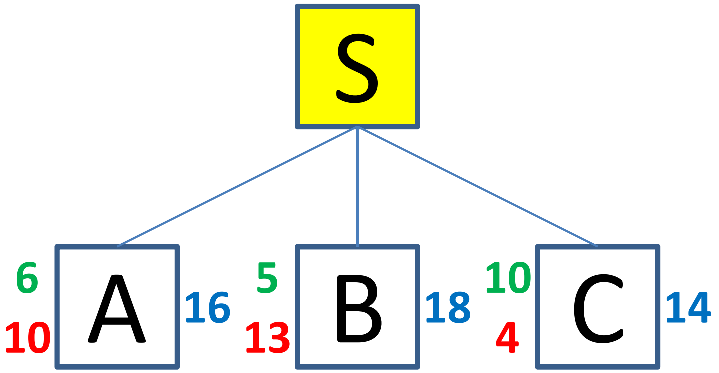


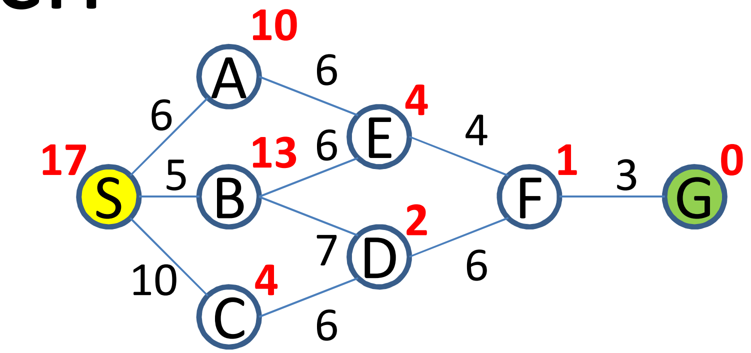


QUEUE:

S

A* Search





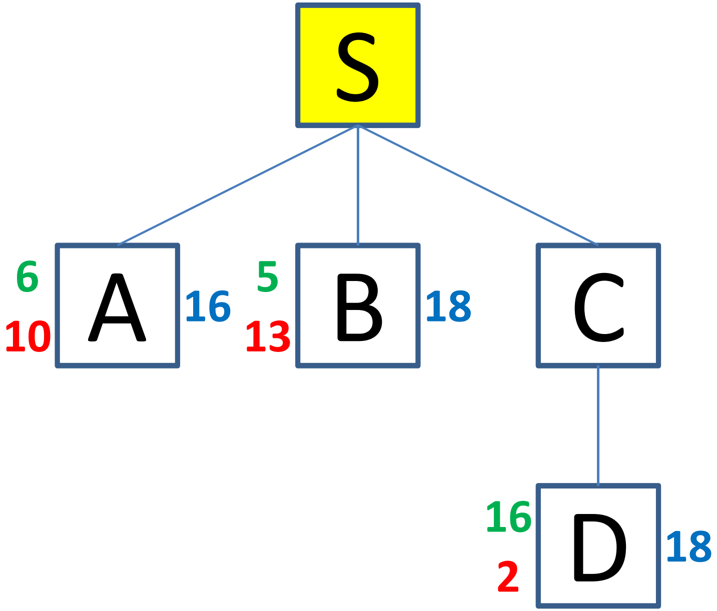
QUEUE:

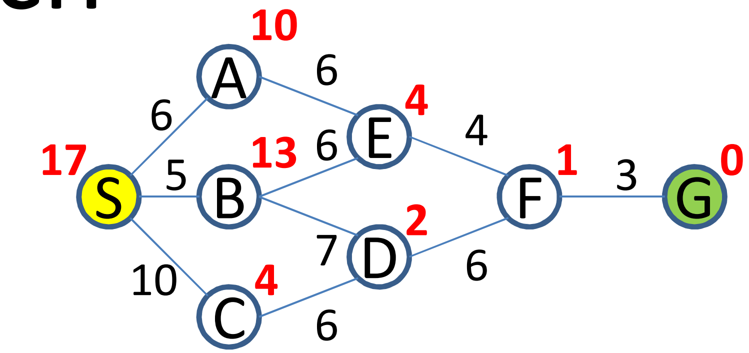
SC

SA

SB

A* Search





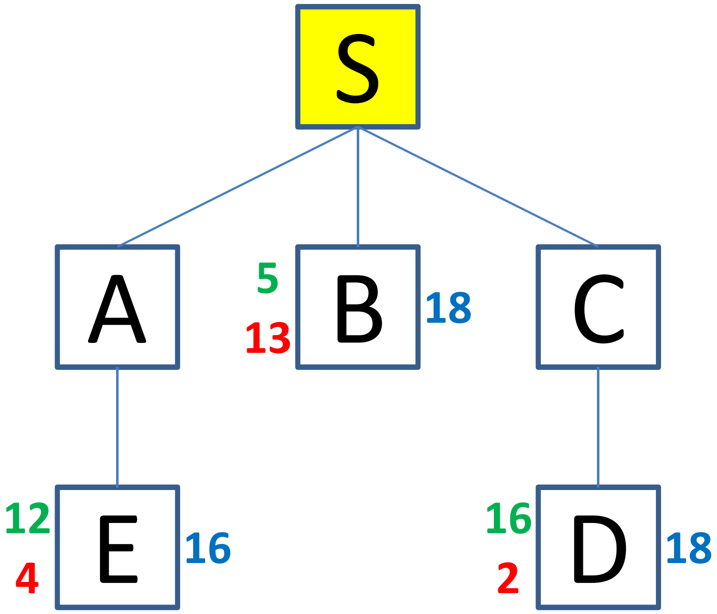
QUEUE:

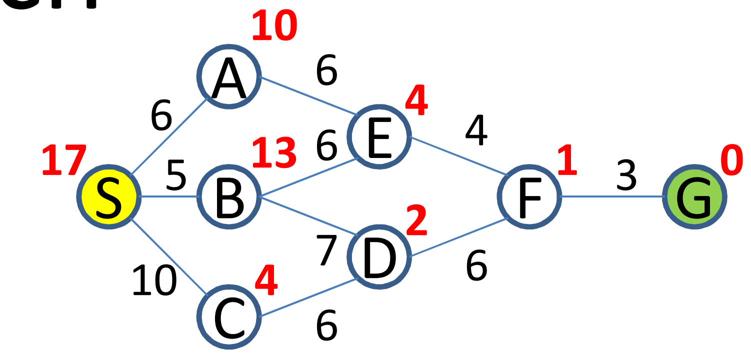
SA

SCD

SB

A* Search





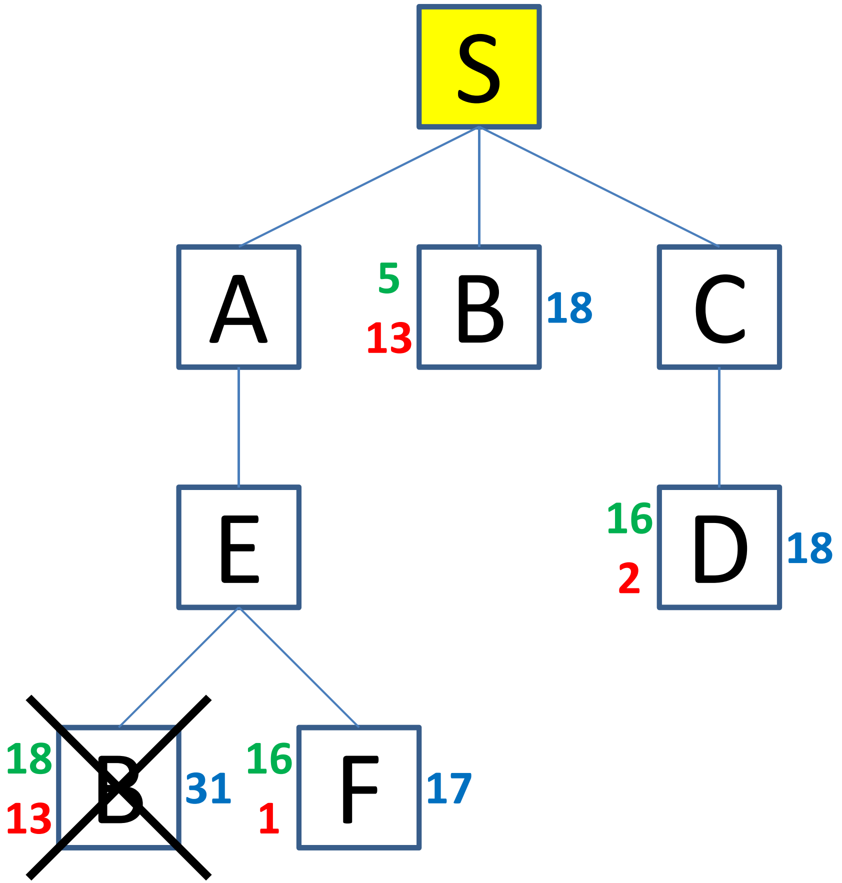
<u>QUEUE:</u>

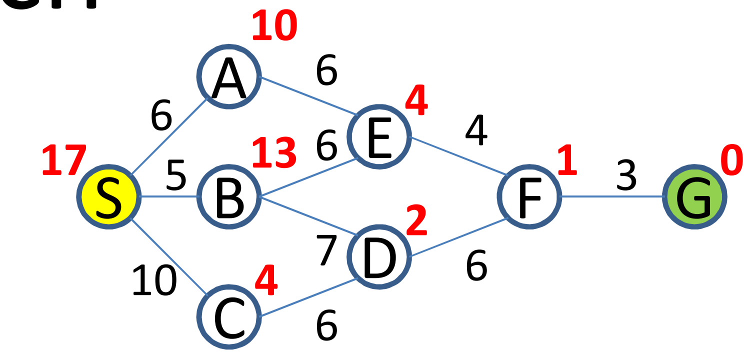
SAE

SCD

SB

A* Search





QUEUE:

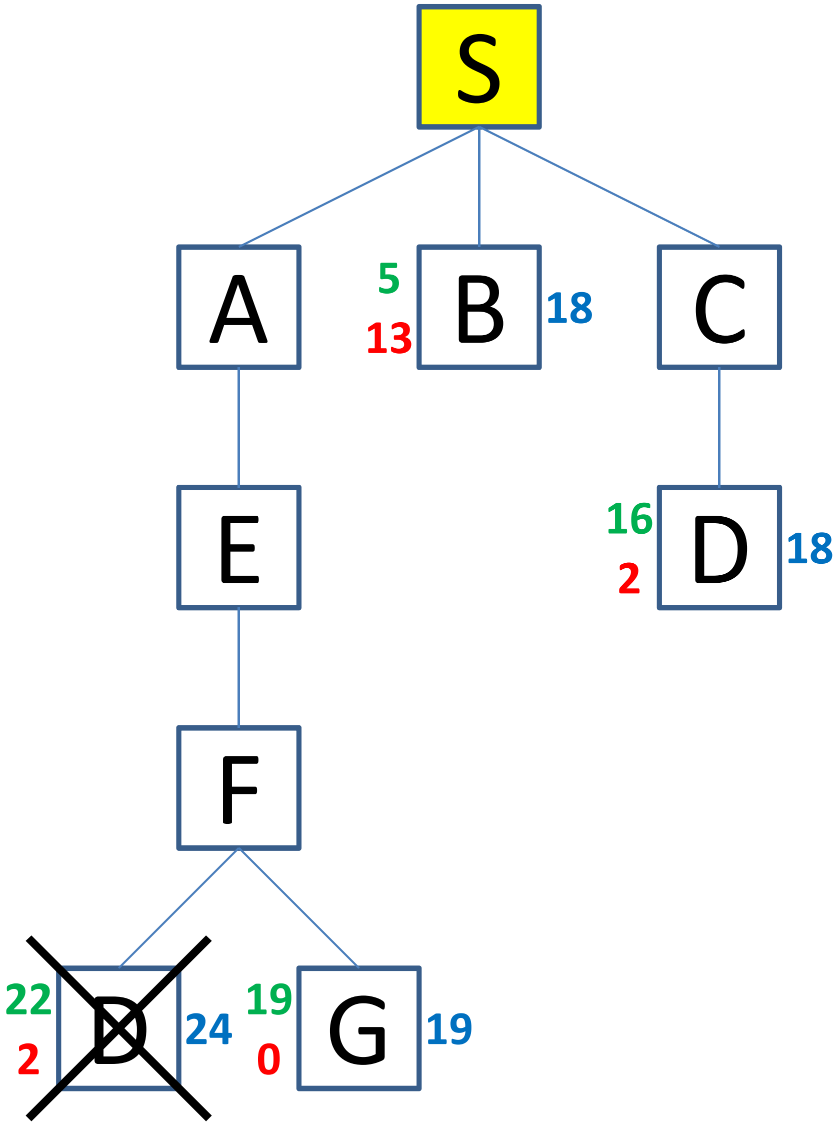
SAEF

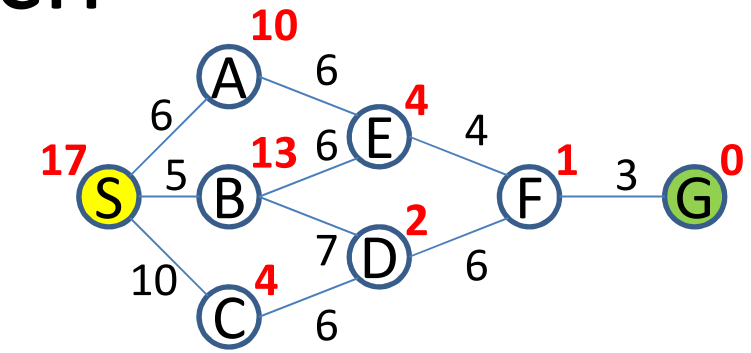
SCD

SB

SAEB

A* Search





QUEUE:

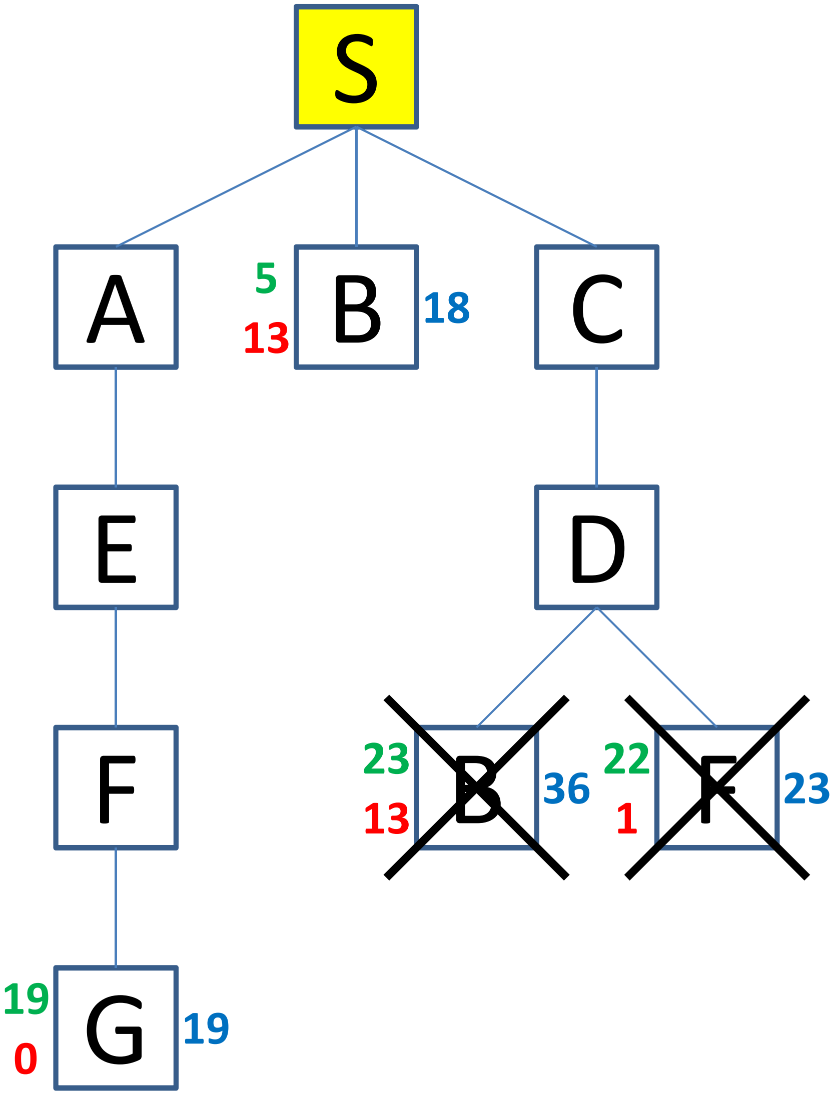
SCD

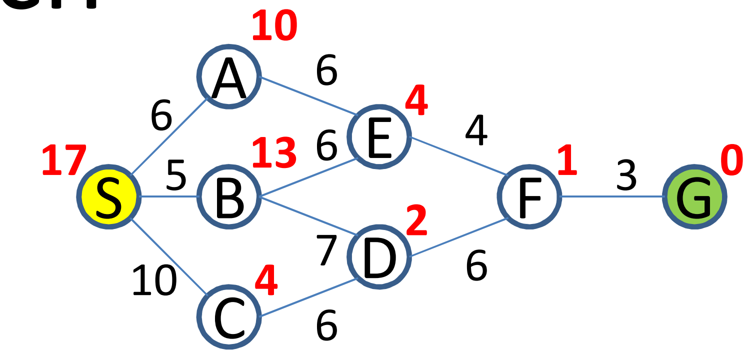
SB

SAEFG

SAEFD

A* Search





QUEUE:

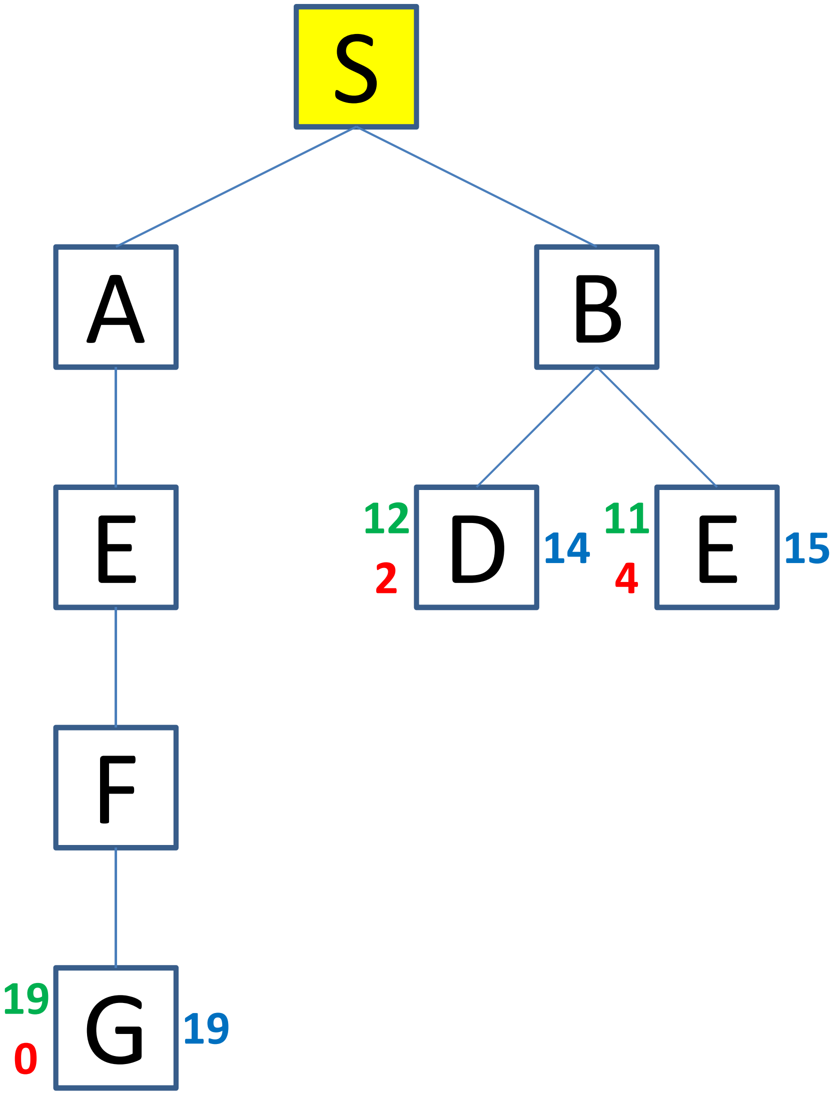
SB

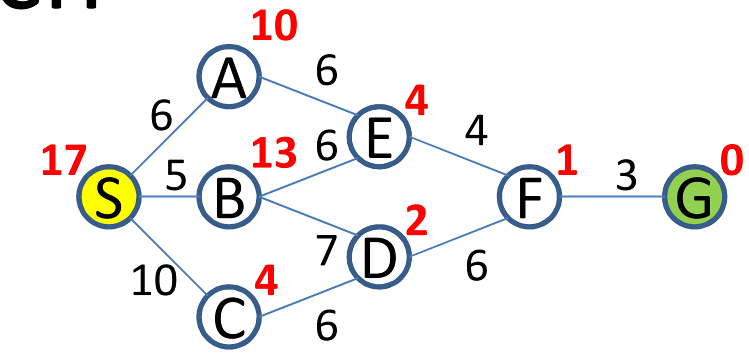
SAEFG

SCDF

SCDB

A* Search





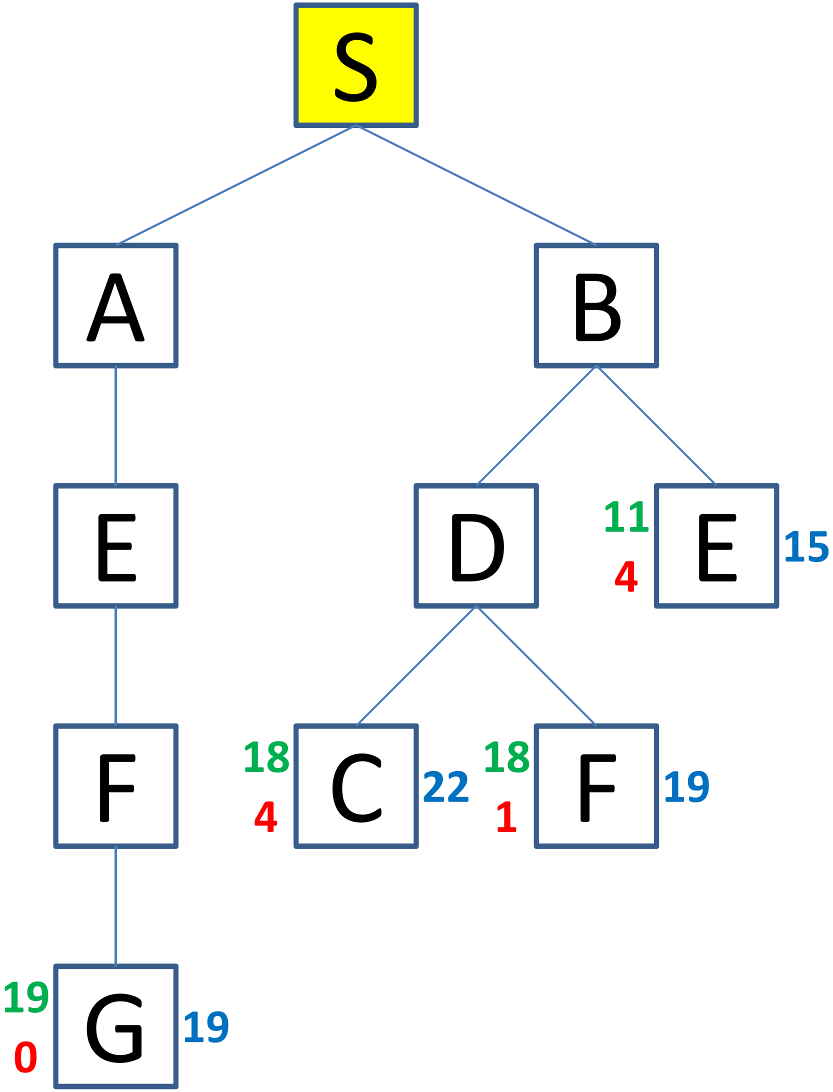
QUEUE:

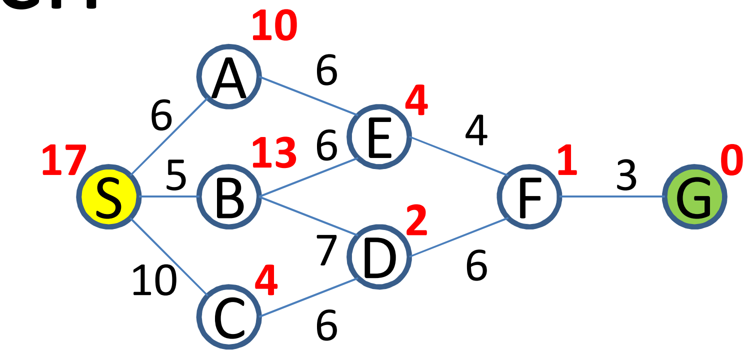
SBD

SBE

SAEFG

A* Search





QUEUE:

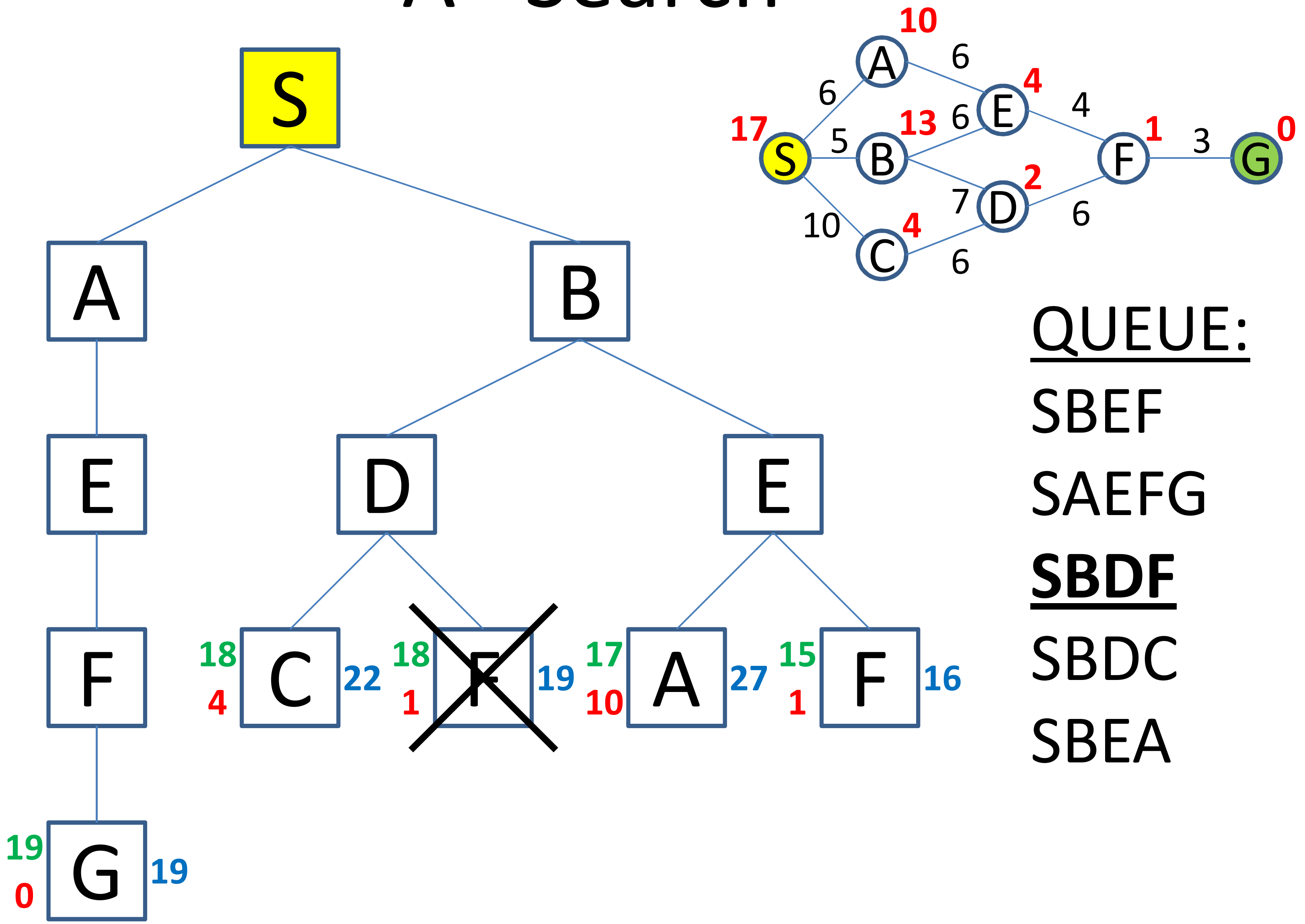
SBE

SBDF

SAEFG

SBDC

A* Search



QUEUE:

SBEF

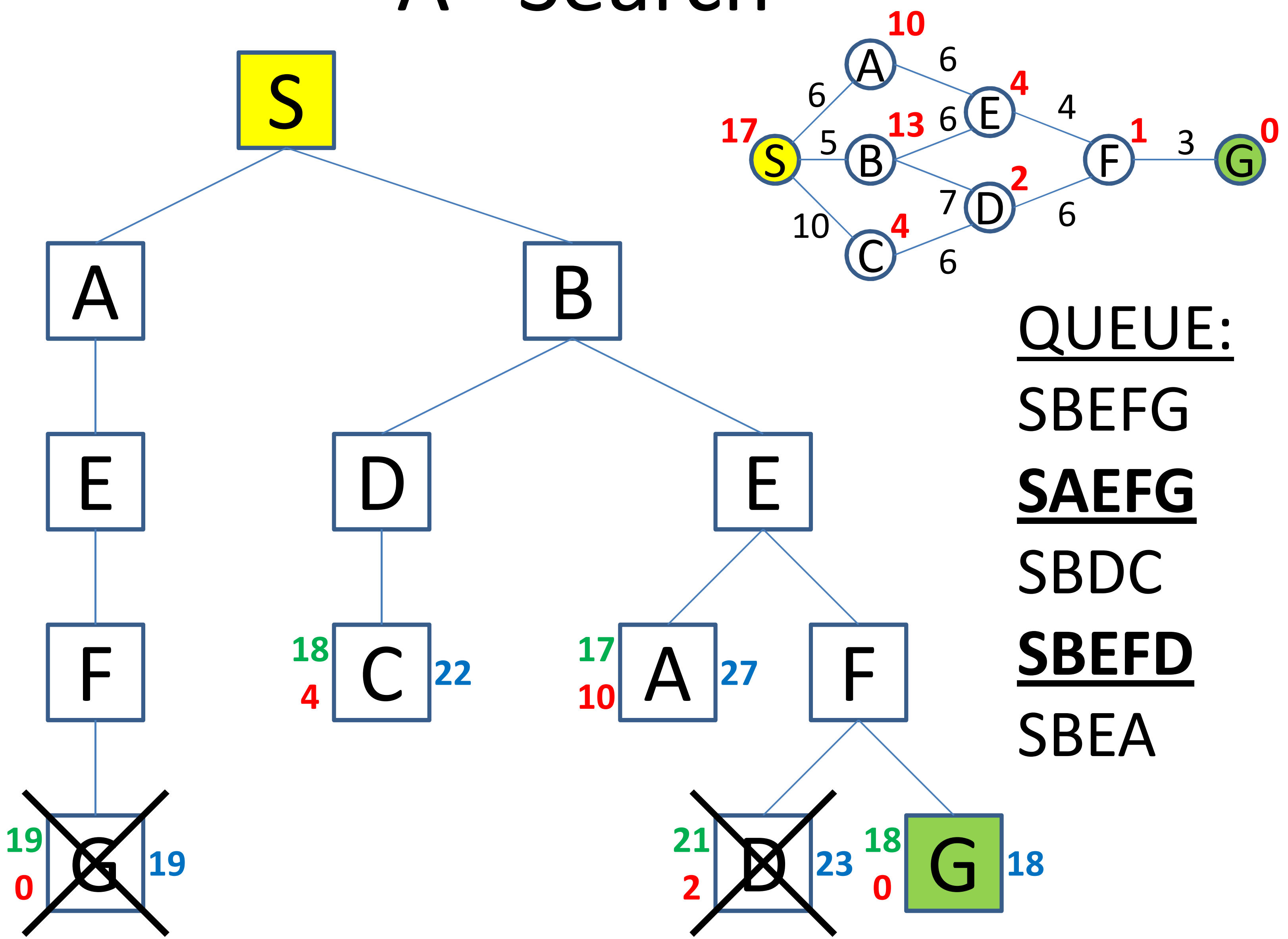
SAEFG

SBDF

SBDC

SBEA

A* Search



QUEUE:

SBEFG

SAEFG

SBDC

SBEFD

SBEA