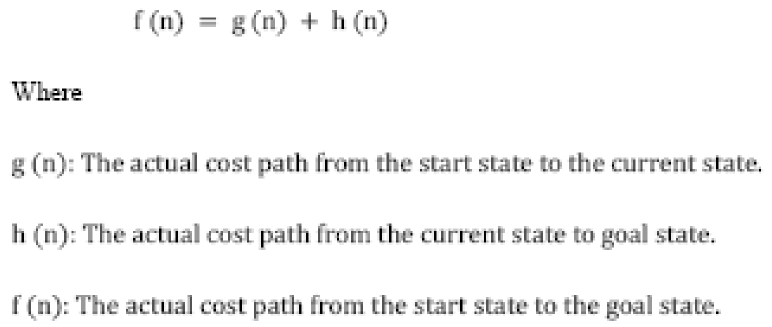
A* Search

- A* is a cornerstone name of many AI systems and has been used since it was developed in 1968 by Peter Hart; Nils Nilsson and Bertram Raphael. It is the combination of Dijkstra's algorithm and Best first search. It can be used to solve many kinds of problems. A* search finds the shortest path through a search space to goal state using heuristic function. This technique finds minimal cost solutions and is directed to a goal state called A* search. In A*, the * is written for optimality purpose. The A* algorithm also finds the lowest cost path between the start and goal state, where changing from one state to another requires some cost. A* requires heuristic function to evaluate the cost of path that passes through the particular state. This algorithm is complete if the branching factor is finite and every action has fixed cost. A* requires heuristic function to evaluate the cost of path that passes through the particular state. It can be defined by following formula.



$$f(n) = g(n) + h(n)$$

Where

$g(n)$: The actual cost path from the start state to the current state.

$h(n)$: The actual cost path from the current state to goal state.

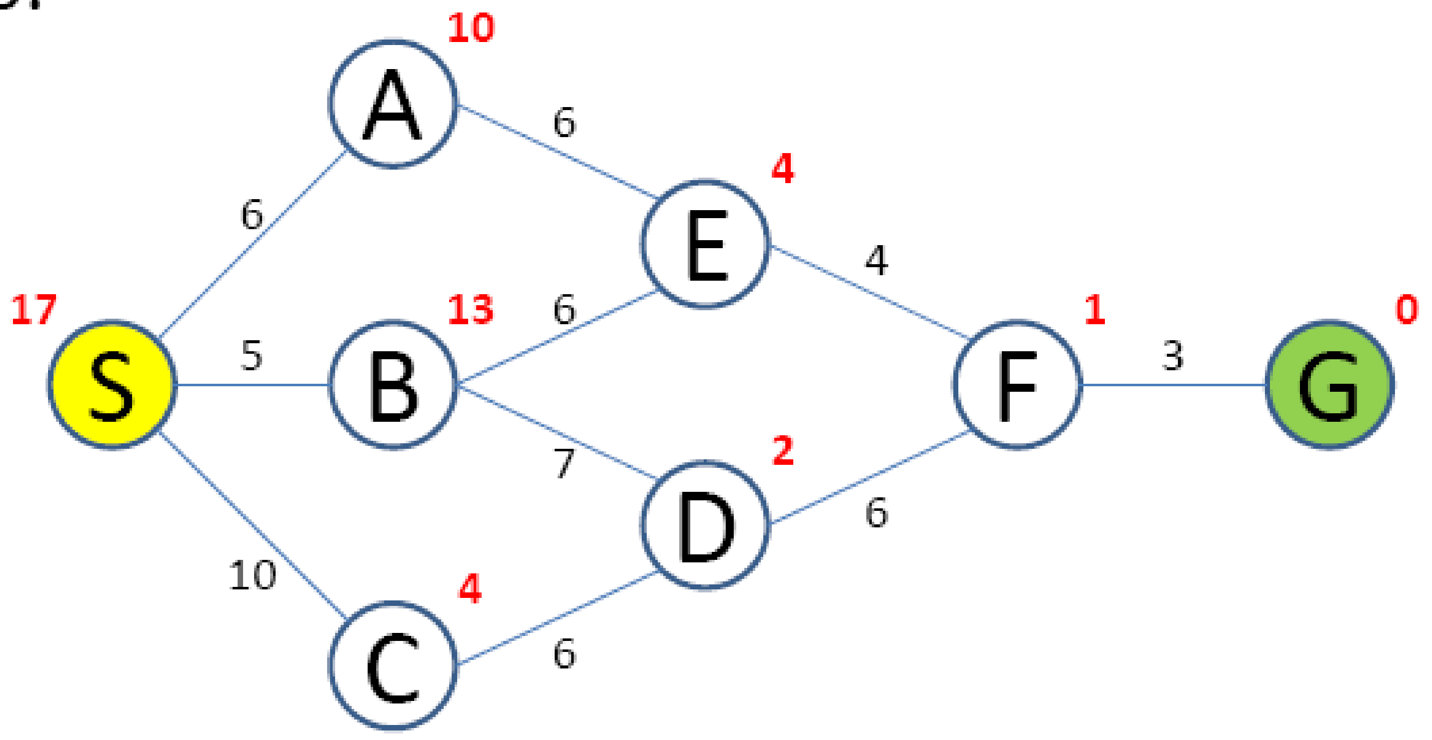
$f(n)$: The actual cost path from the start state to the goal state.

A* Algorithm

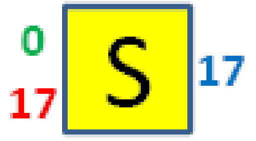
```
1.  Initialize: set OPEN=[s], CLOSED=[], $g(s)=0$, $f(s)=h(s)$
2.  Fail: If OPEN=[], then terminate and fail
3.  Select: Select a state with minimum cost ,n, from OPEN and save in
    CLOSED
4.  Terminate: If $n \in G$ then terminate with success and return $f(s)$
5.  Expand: For each successors , m of n
    For each successor, m, insert m in OPEN only if
          if $m \notin [OPEN \cup CLOSED]$
     set $g(m)= g[n]+C[n,m]$
    Set $f(m)=g(m)+h(n)$
    if $m \in [OPEN \cup CLOSED]$
    set $g(m)= \min\{g[m], g(n)+C[n,m]\}$
    Set $f(m)=g(m)+h(m)$
    If $f[m]$ has decreased and $m \in CLOSED$ move m to OPEN
6.  Loop: Goto step 2
```

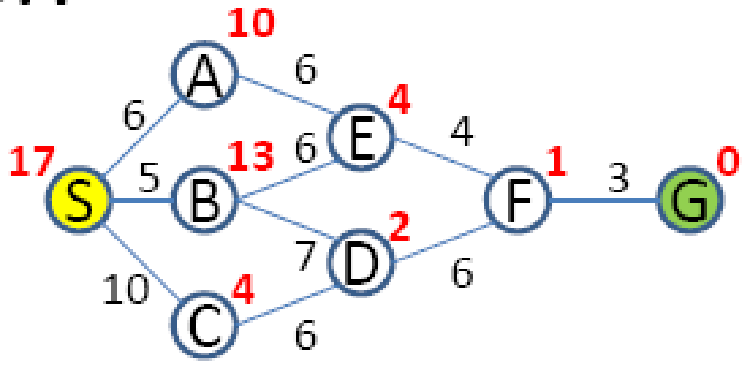
Problem

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



A* Search

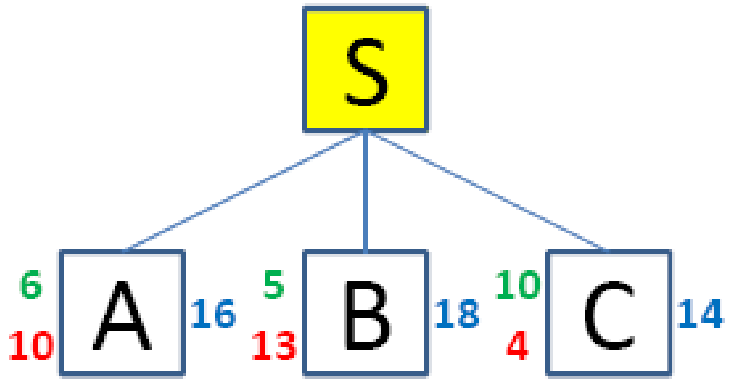


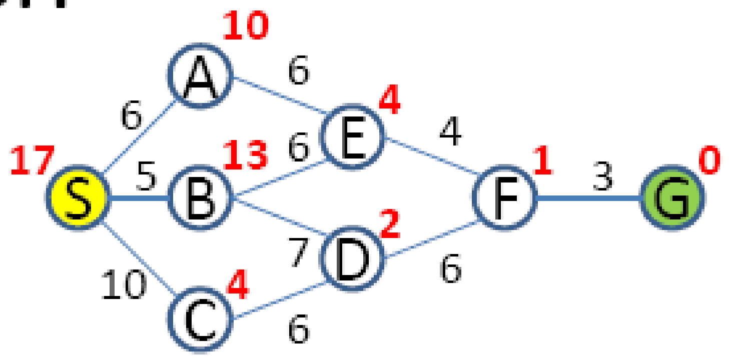


QUEUE:

S

A* Search





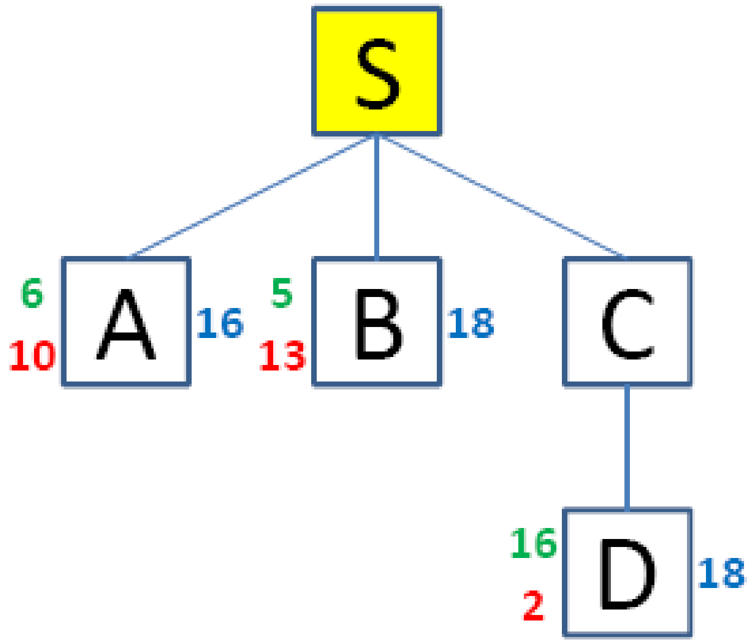
QUEUE:

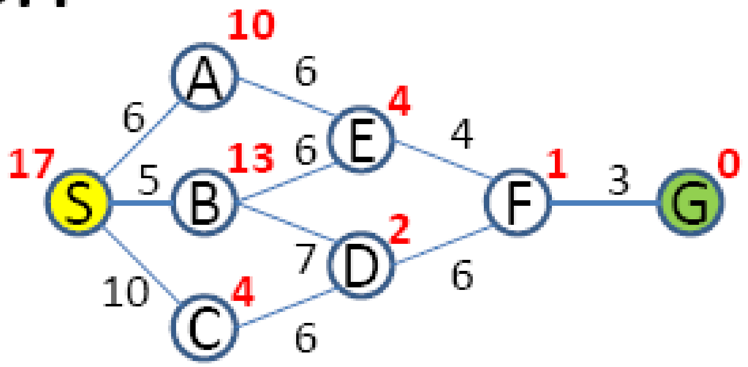
SC

SA

SB

A* Search





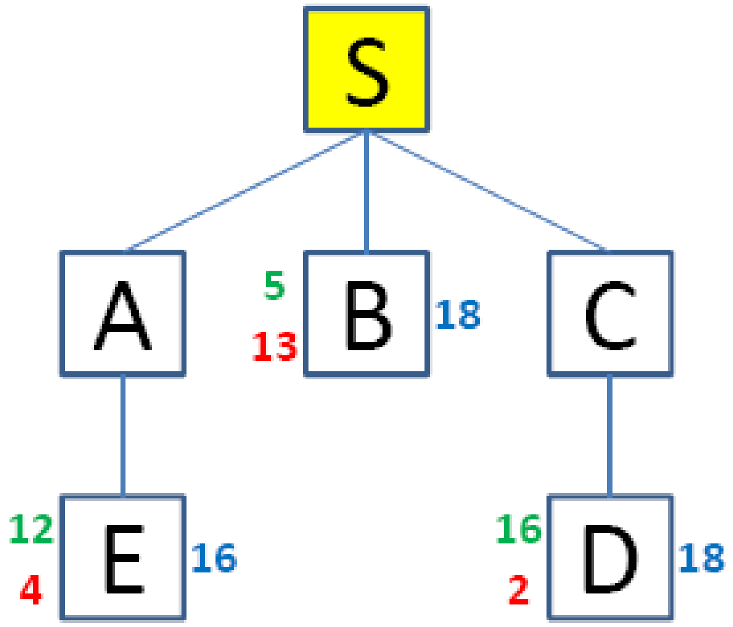
QUEUE:

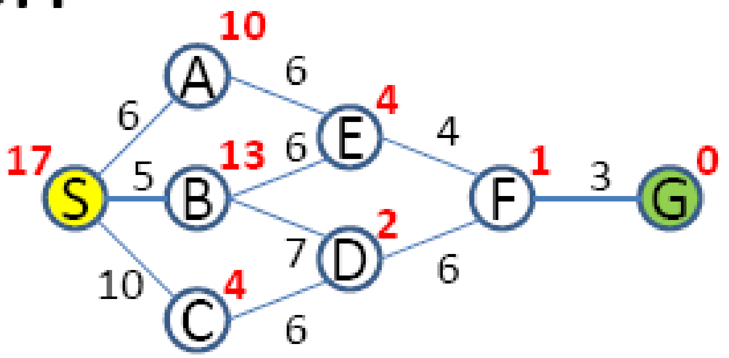
SA

SCD

SB

A* Search





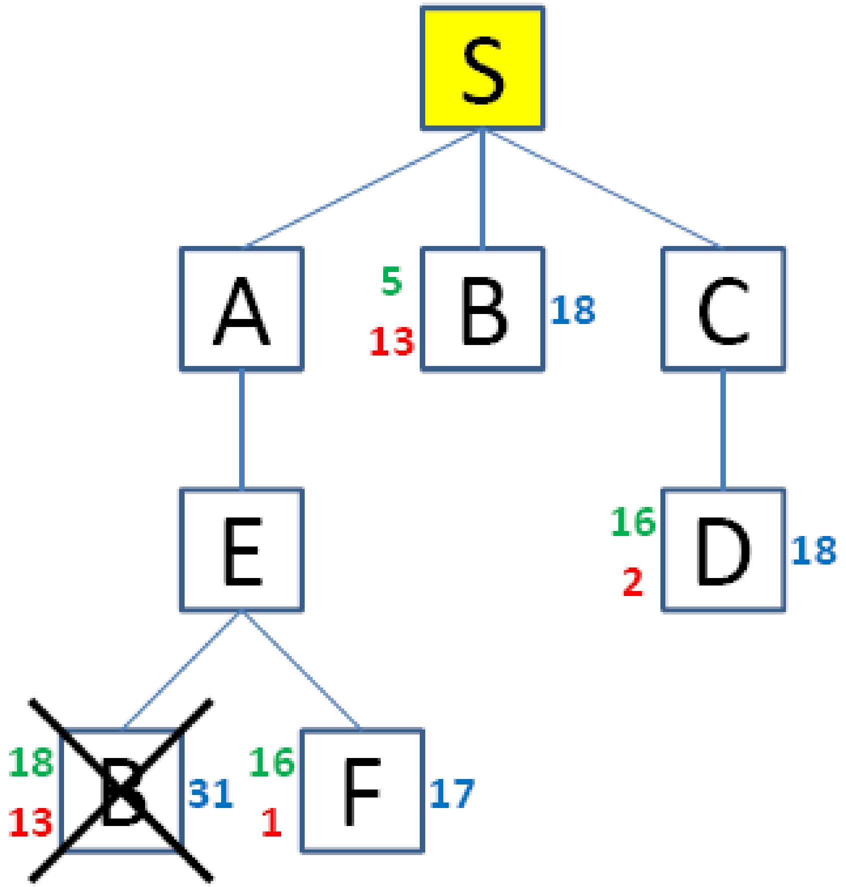
QUEUE:

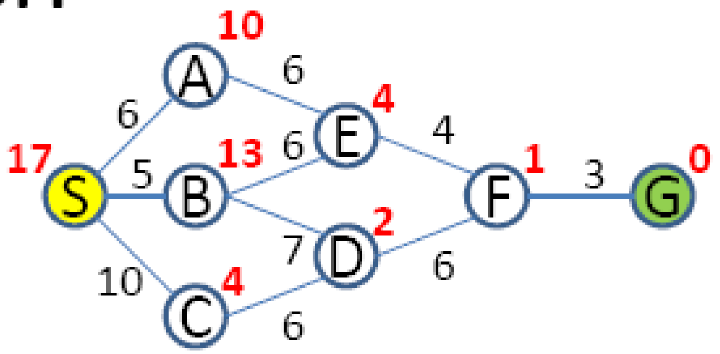
SAE

SCD

SB

A* Search





QUEUE:

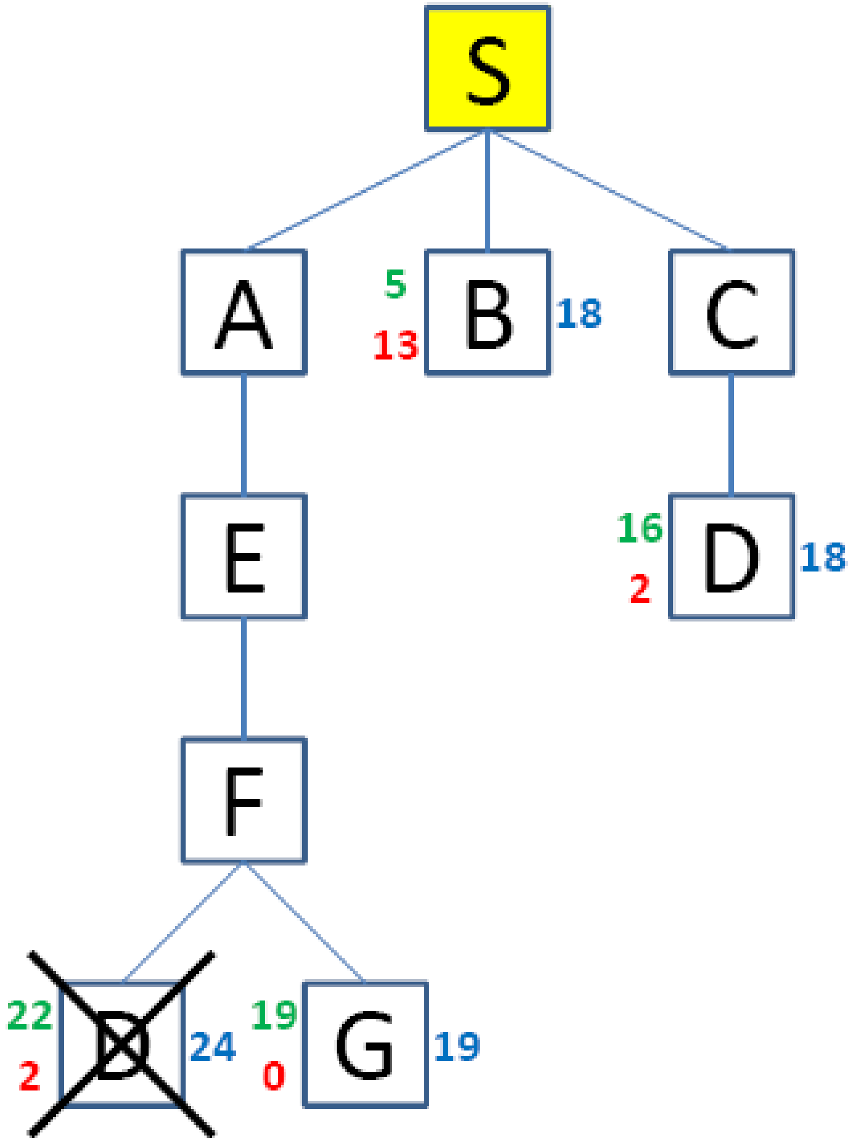
SAEF

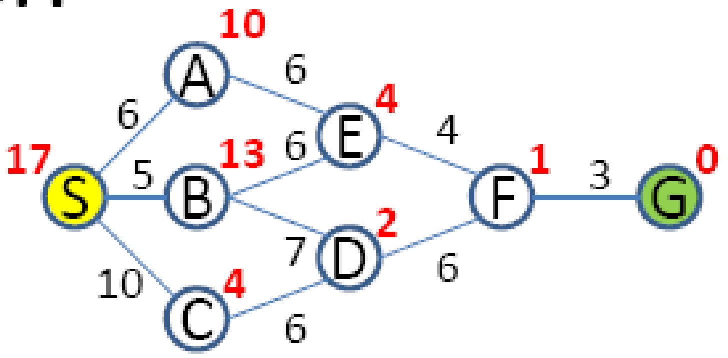
SCD

SB

SAEB

A* Search





QUEUE:

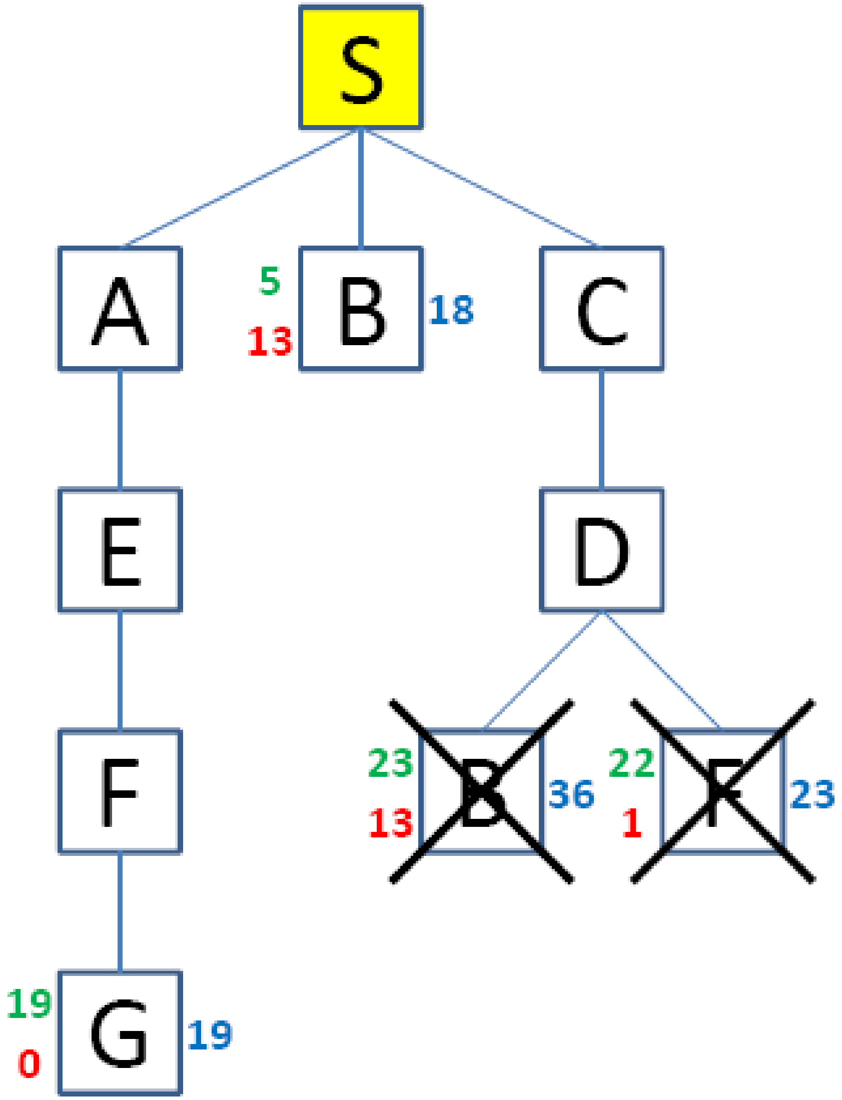
SCD

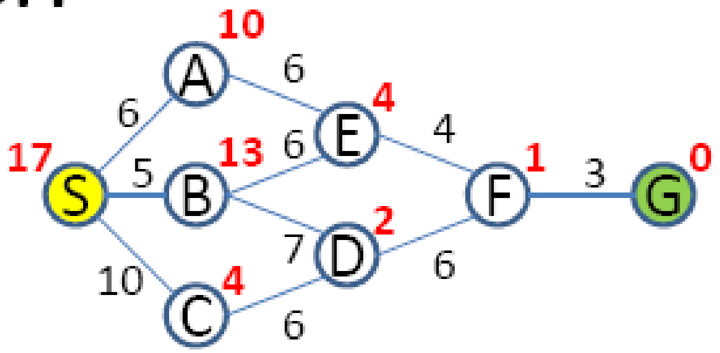
SB

SAEFG

SAEFD

A* Search





<u>QUEUE:</u>

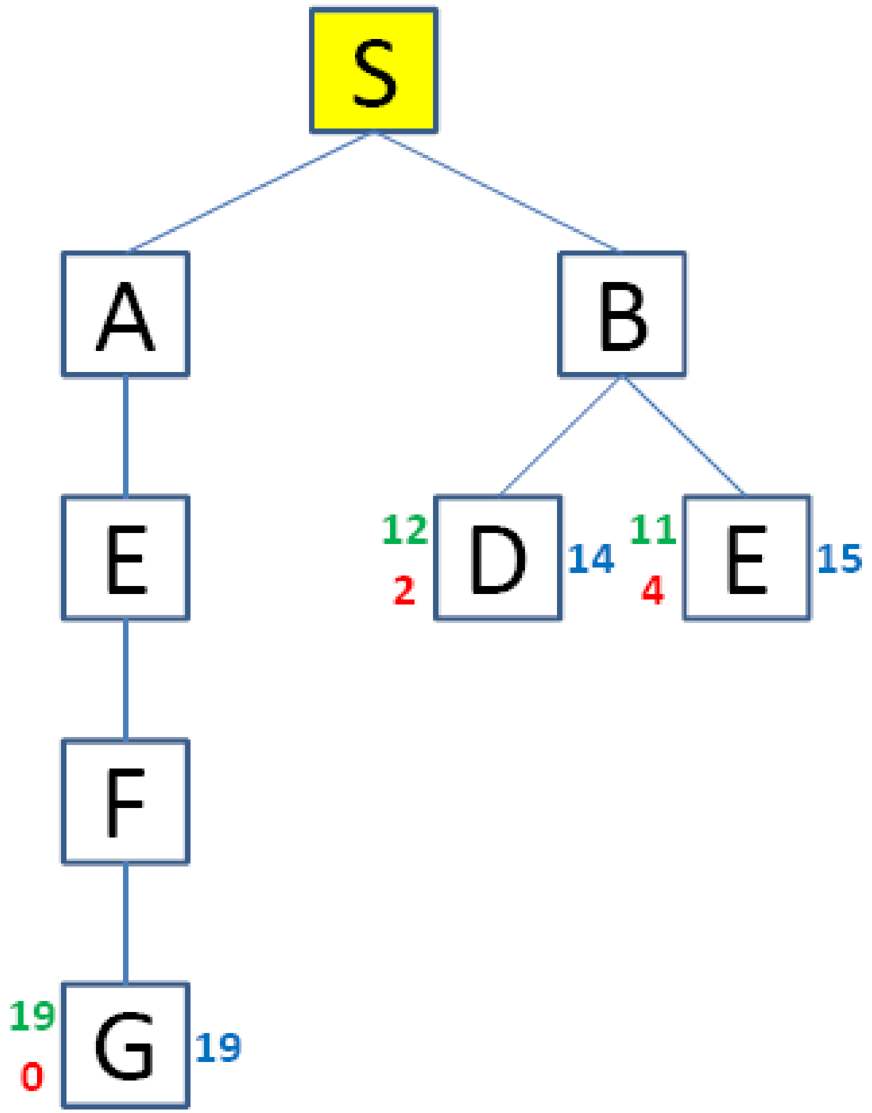
SB

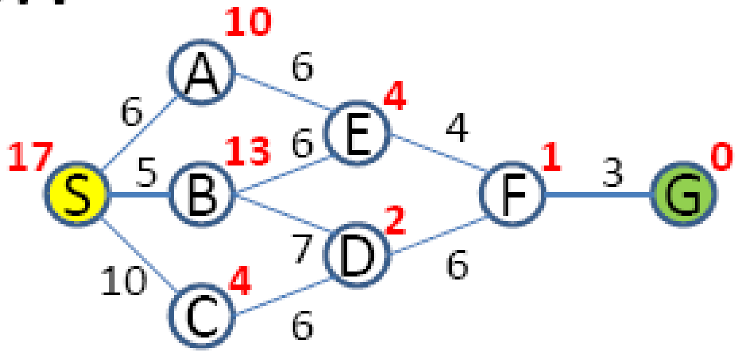
SAEFG

<u>SCDF</u>

<u>SCDB</u>

A* Search





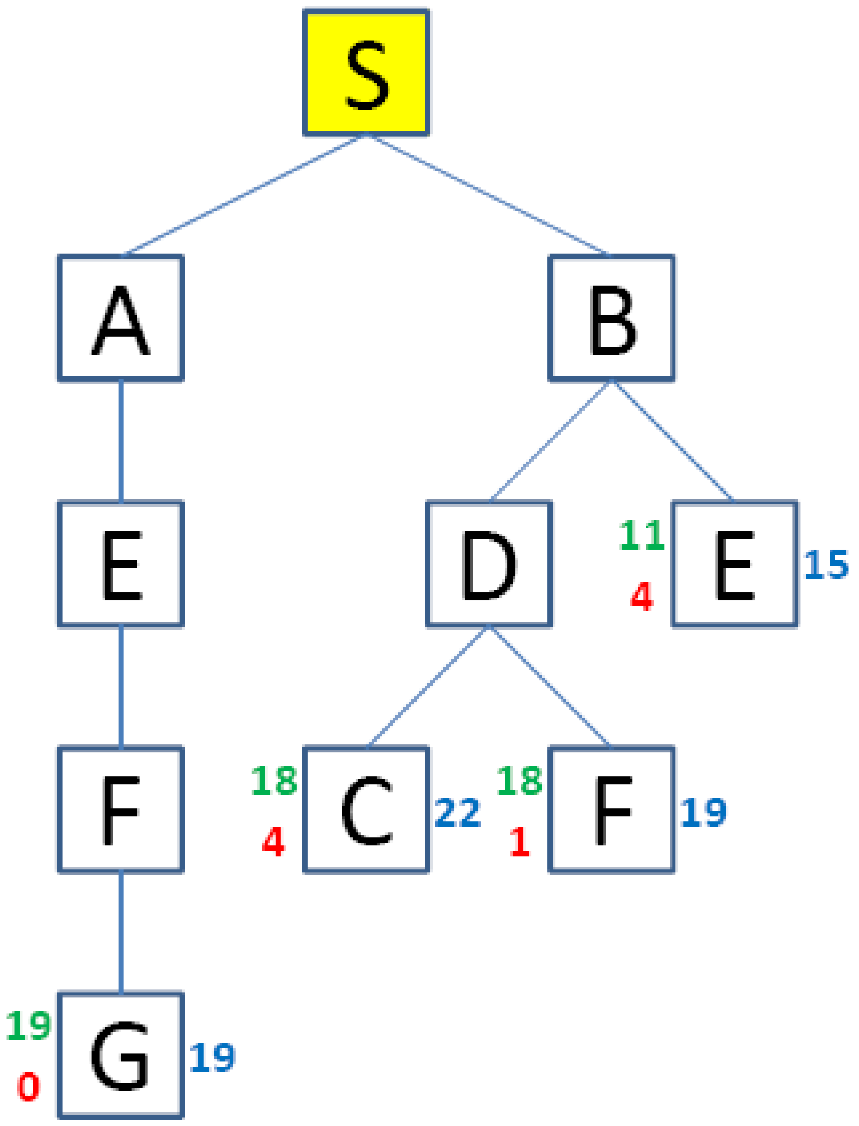
QUEUE:

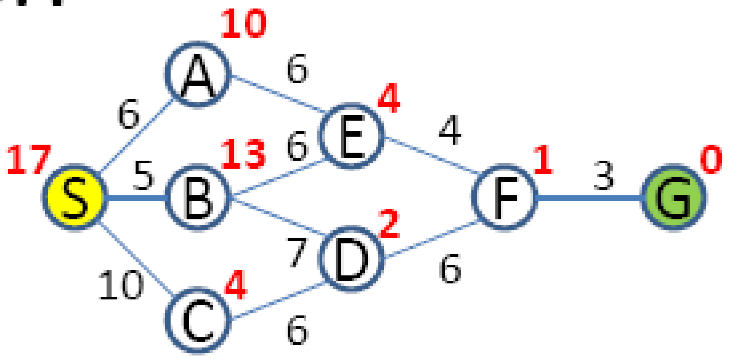
SBD

SBE

SAEFG

A* Search





QUEUE:

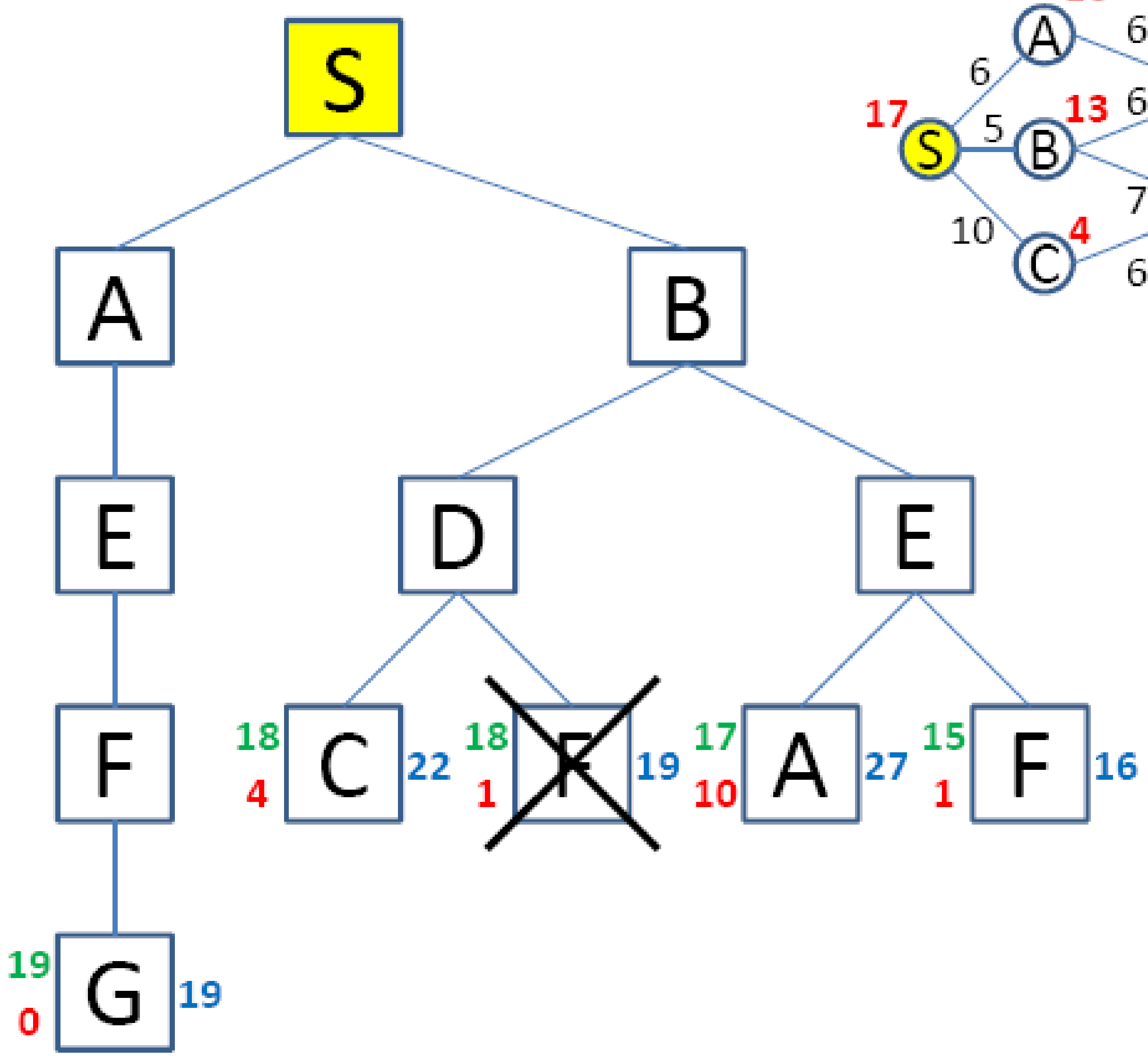
SBE

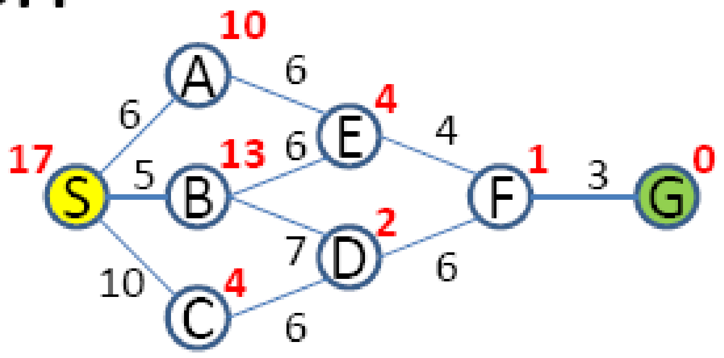
SBDF

SAEFG

SBDC

A* Search





QUEUE:

SBEF

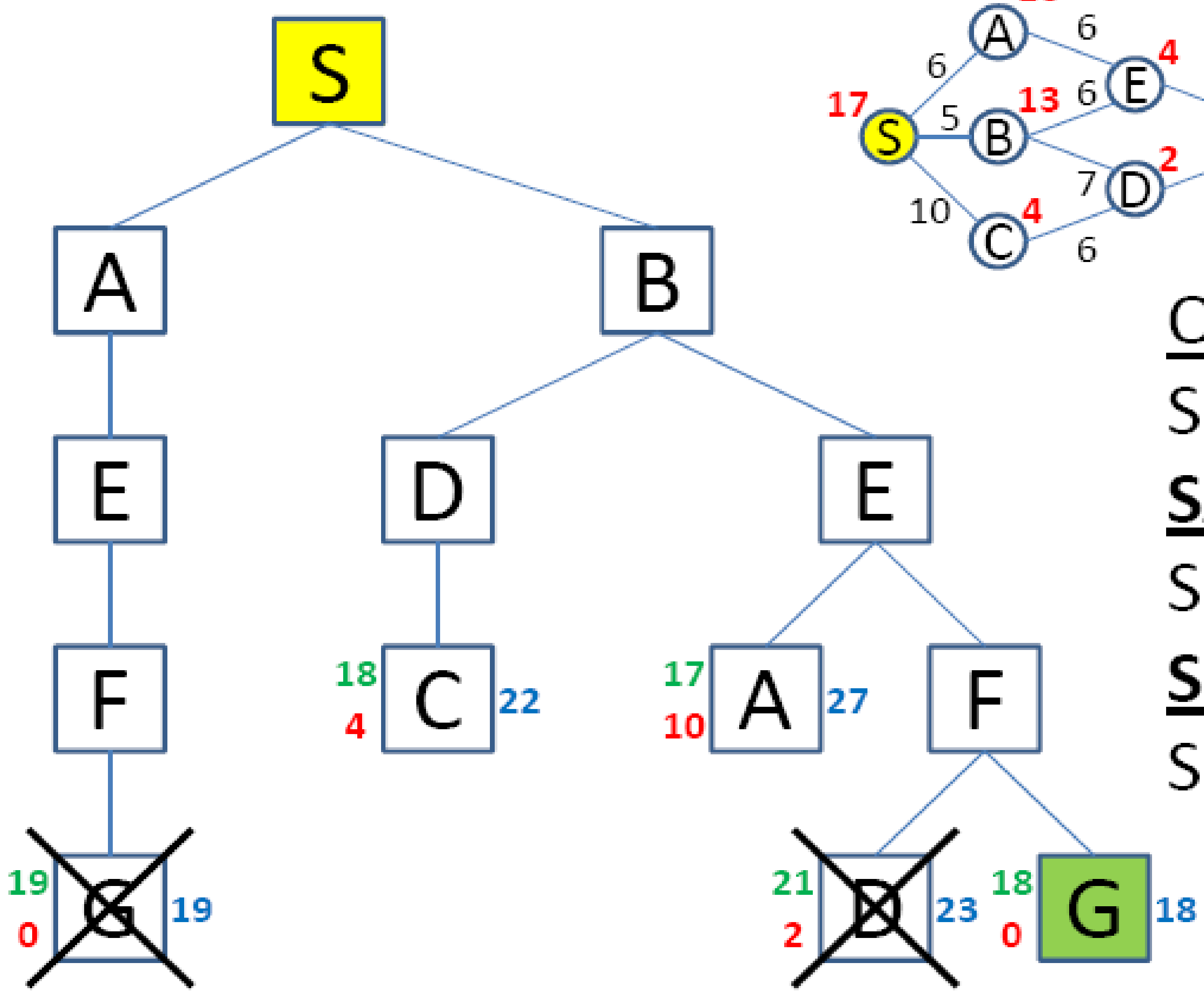
SAEFG

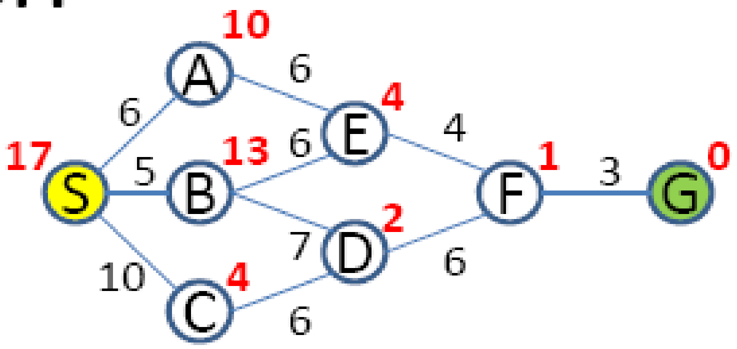
SBDF

SBDC

SBEA

A* Search





QUEUE:

SBEFG

SAEFG

SBDC

SBEFD

SBEA