ARTIFICIAL INTELLIGENCE VBDS1402

Module 1

Chapter 2

Informed Search (IDA*: Memory bounded search)

IN THIS SESSION YOU WILL LEARN:

IDA* Search (How it works?)

IDA* Example

IDA* algorithm

Advantages and Disadvantages of IDA*

IDA* SEARCH (HOW IT WORKS?)

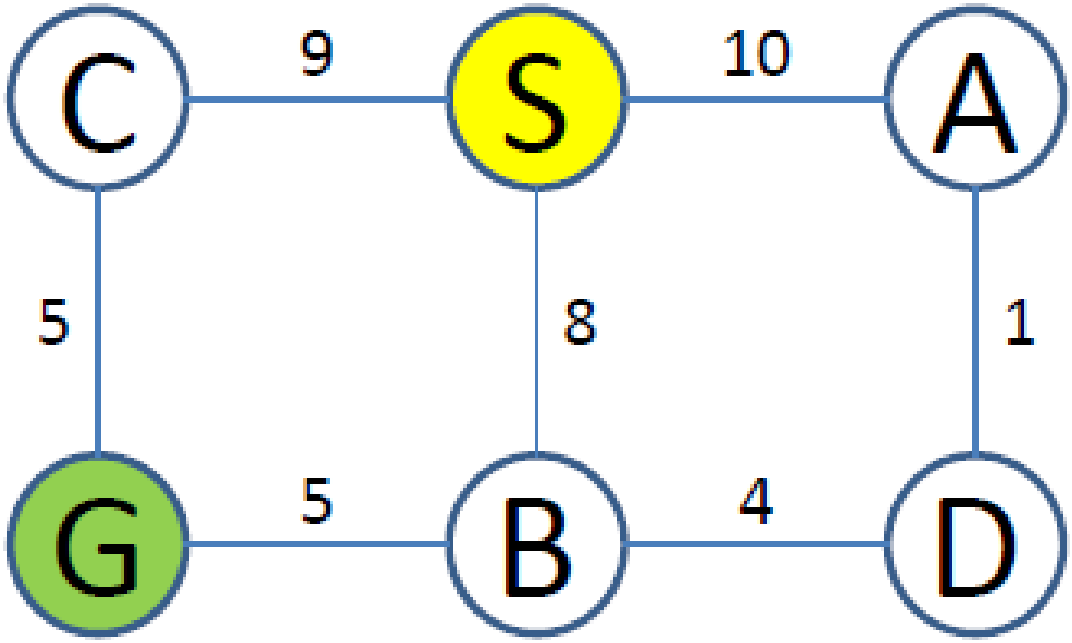
A cost threshold is set.

$f(n) = g(n) + h(n)$ is computed in each iteration.

If $f(n) <$ threshold we expand the node.

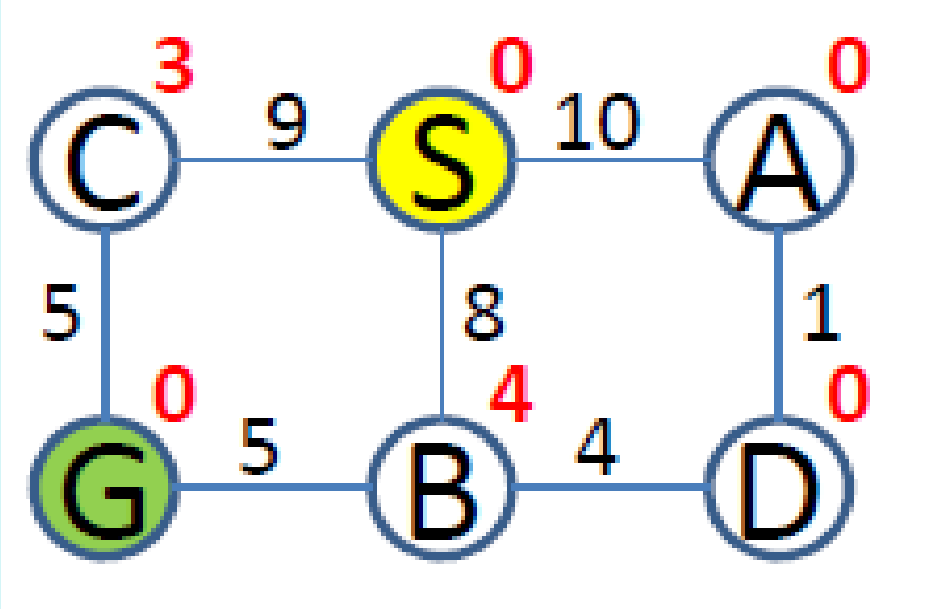
Else the branch is pruned (we don't expand it).

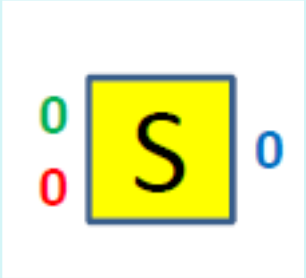
IDA* ALGORITHM



	S	A	B	C	D	G
heuristic	0	0	4	3	0	0

IDA* SEARCH



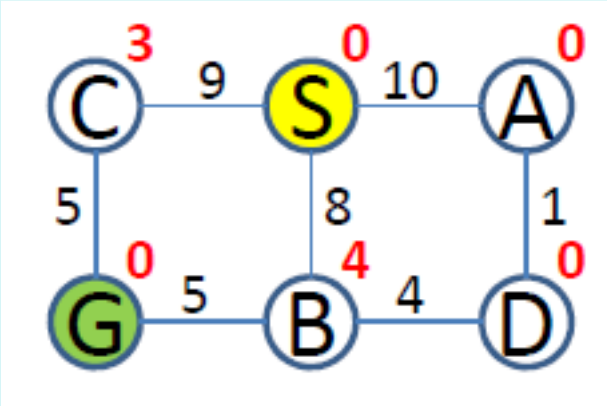


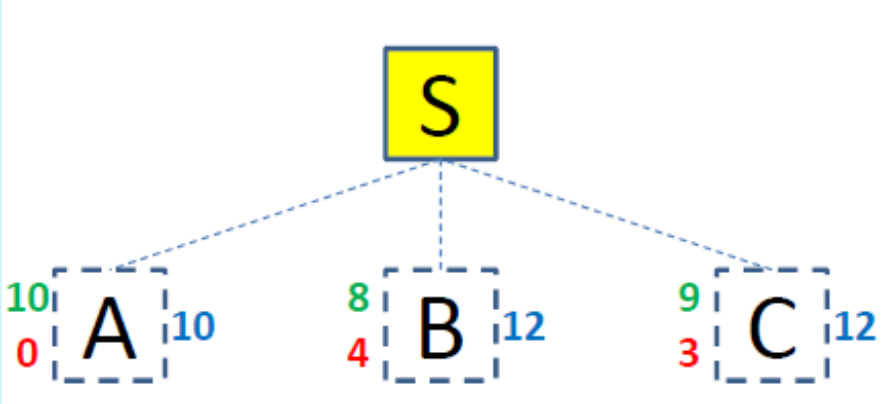
F-bound=0

(threshold)

F-new=∞

IDA* SEARCH





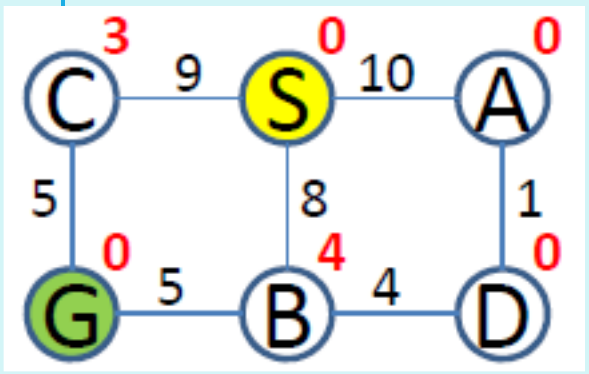
F-bound=0

F-new=10

Children are explored using depth-first

* Dotted lines means pruned nodes. Here, all values are greater than 0, so can't expand.

IDA* SEARCH

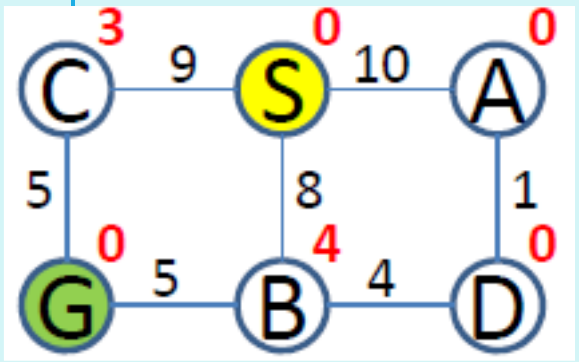


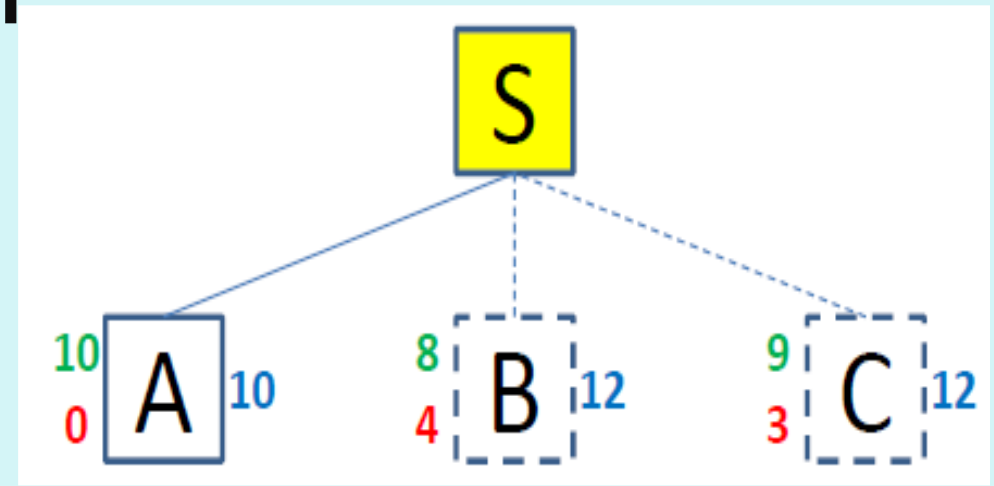
F-bound=10

F-new=∞

* Reset the threshold to minimum value of the node.

IDA* SEARCH



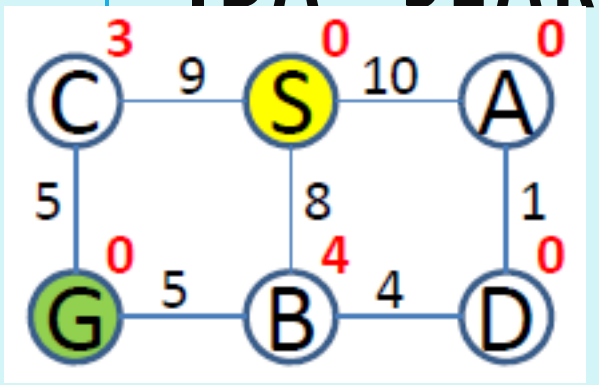


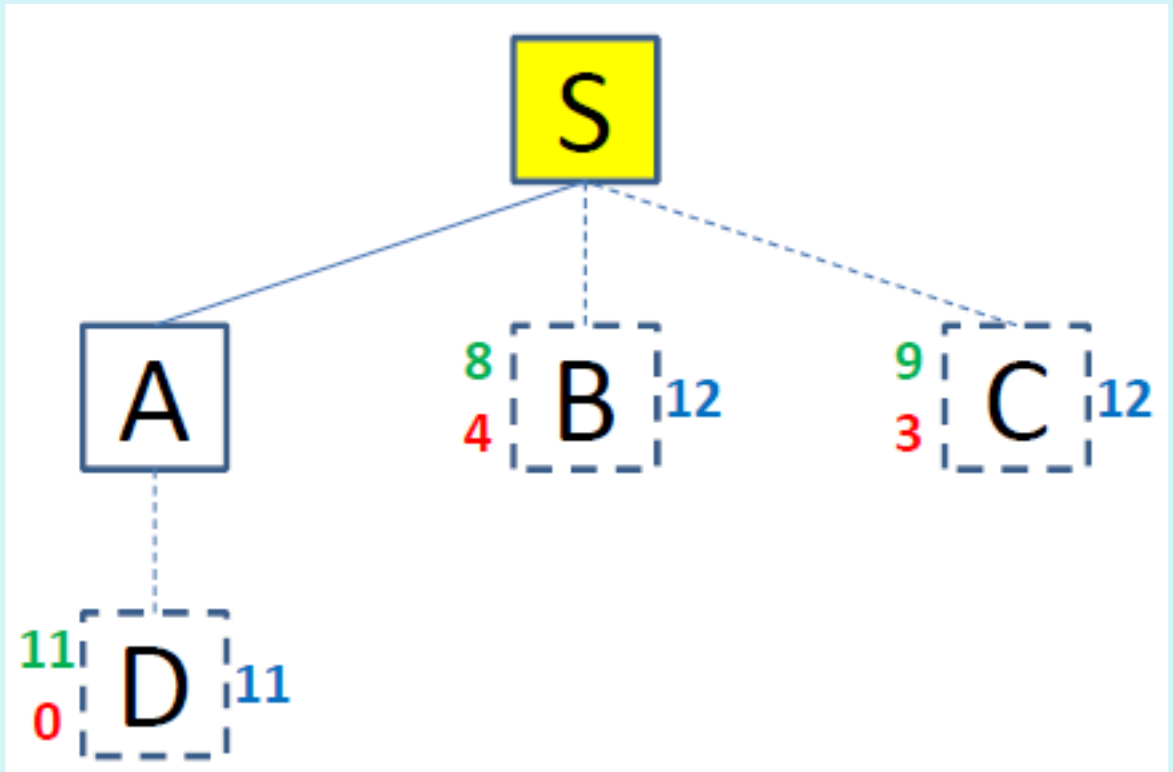
F-bound=10

F-new=12

* Now, the threshold is 10, hence, A can be expanded. You can see thick line for A. But the cost with B and C is more than 10, hence, pruned.

IDA* SEARCH



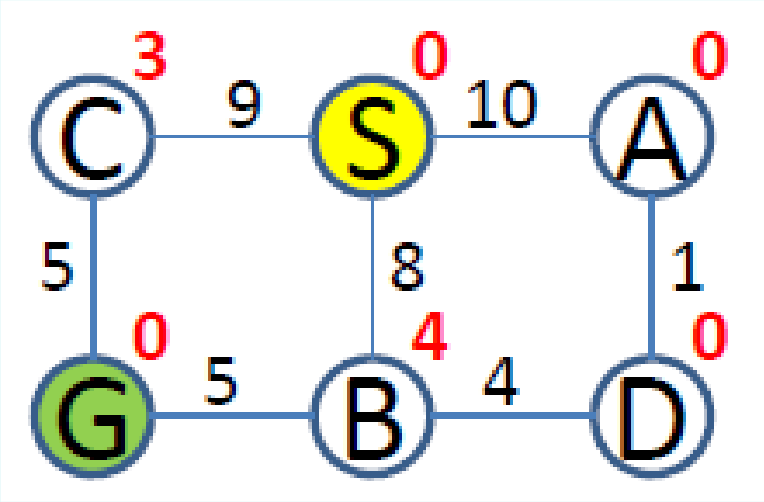


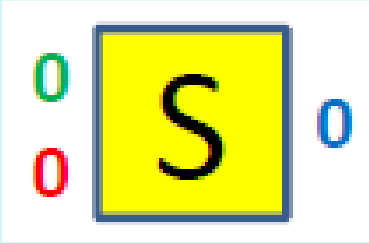
F-bound=10

F-new=11

* Expand A

IDA* SEARCH

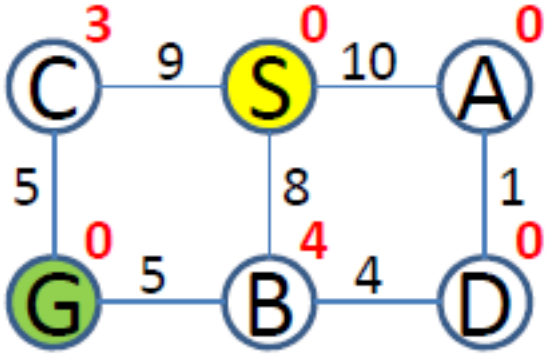


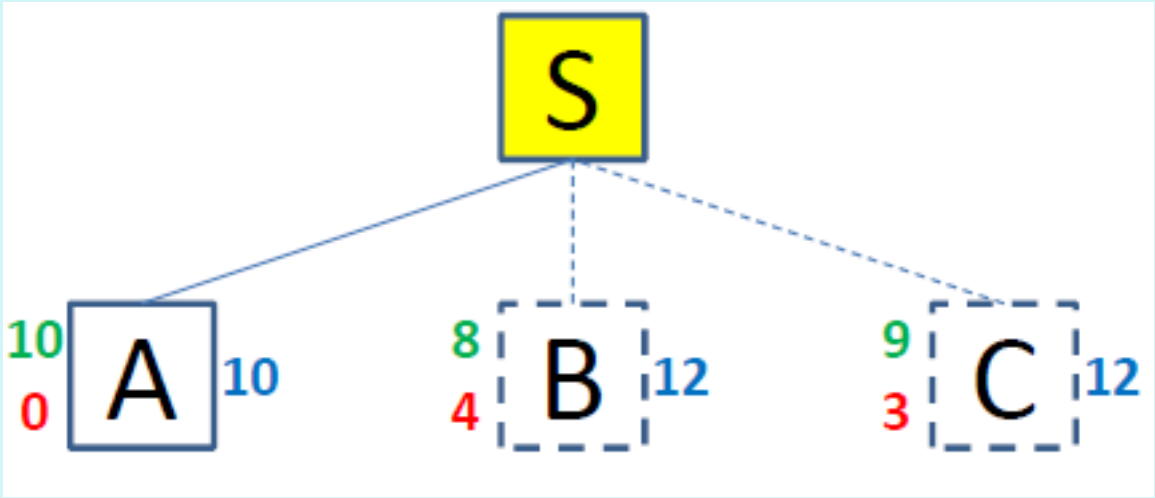


F-bound=11

F-new=∞

IDA* SEARCH

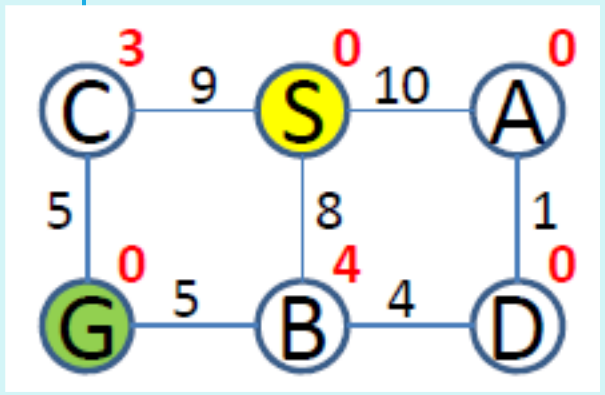


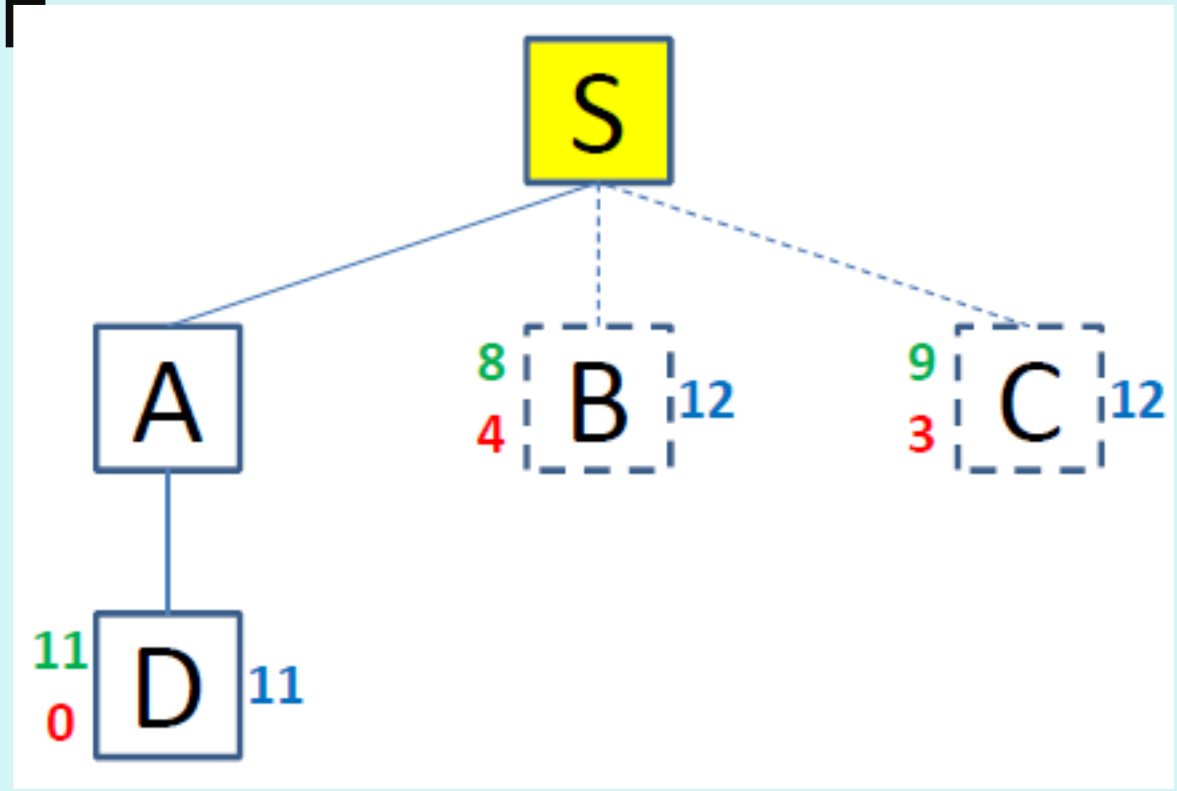


F-bound=11

F-new=12

IDA* SEARCH

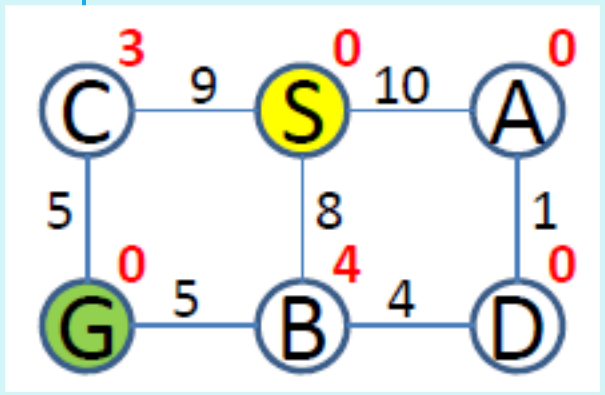


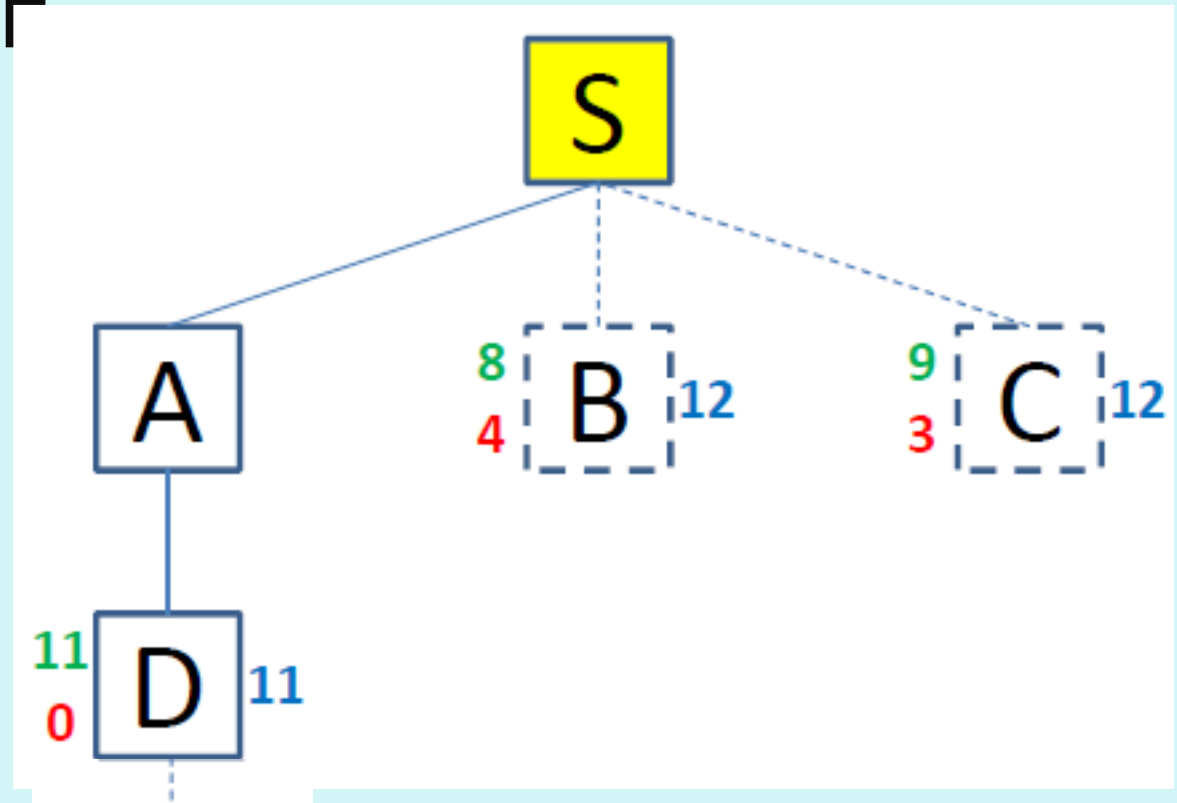


F-bound=11

F-new=12

IDA* SEARCH

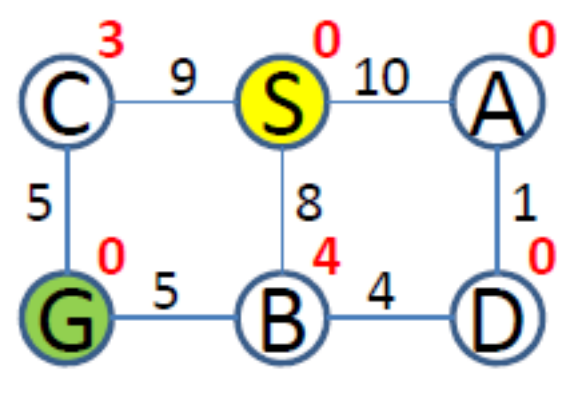




F-bound=11

F-new=12

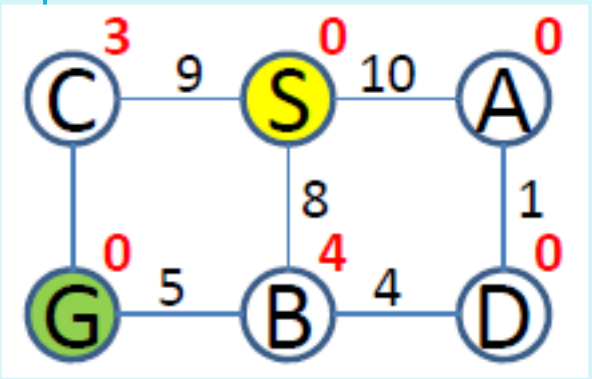
IDA* SEARCH

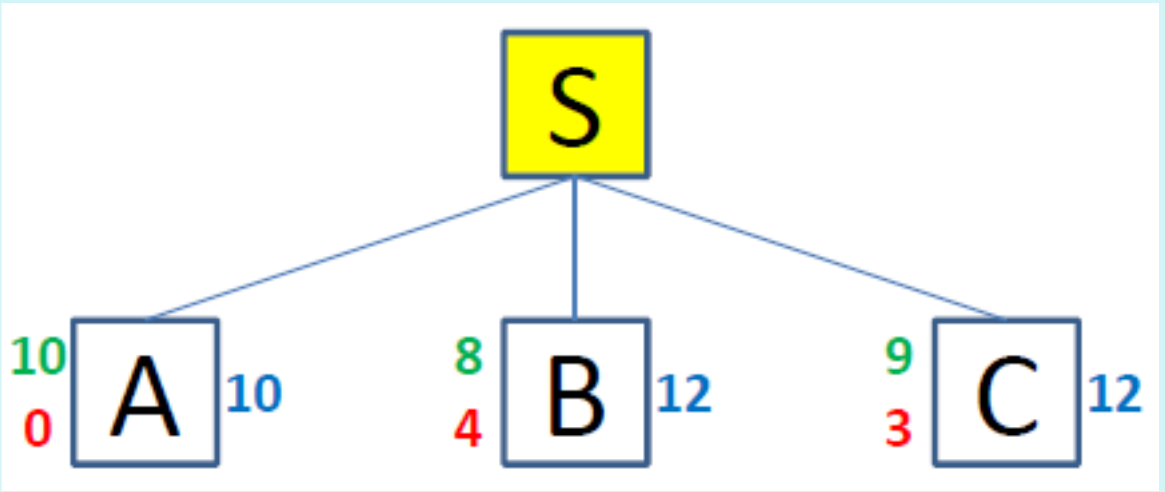


F-bound=12

F-new=∞

IDA* SEARCH

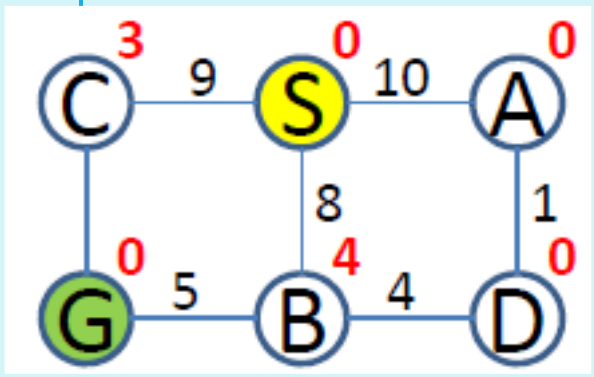


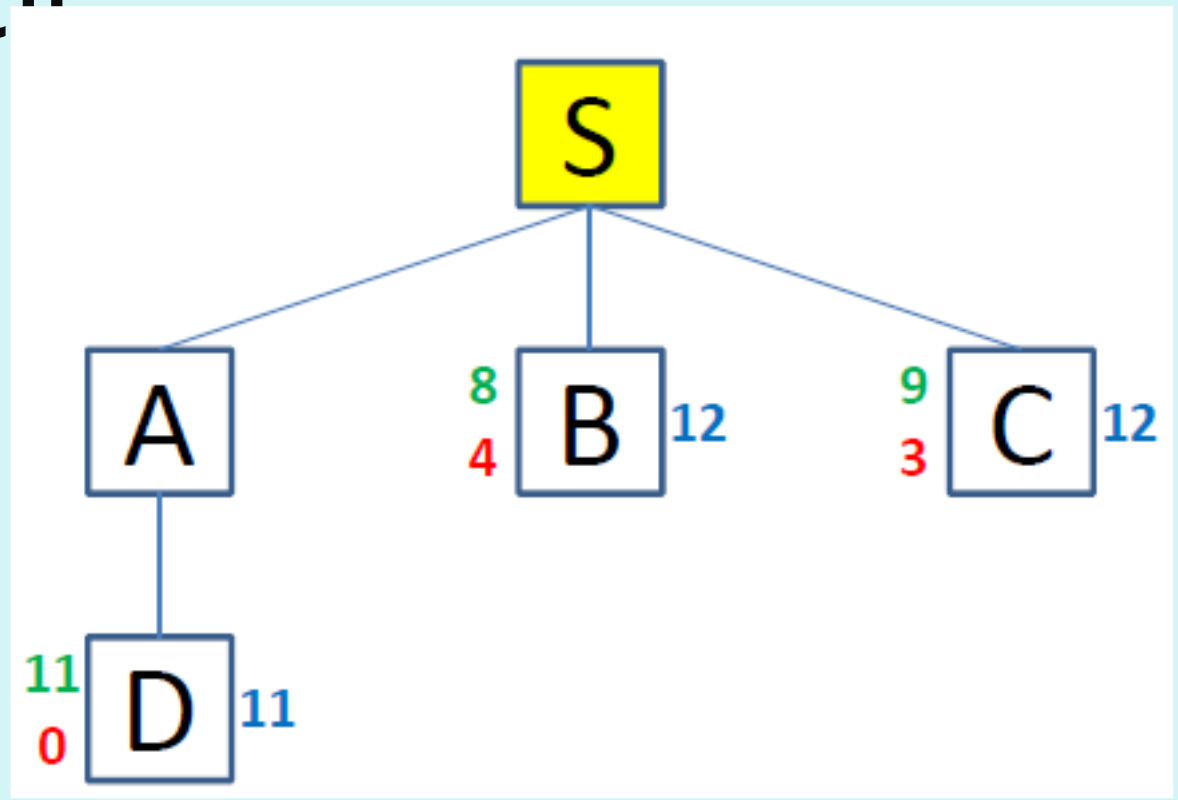


F-bound=12

F-new=∞

IDA* SEARCH

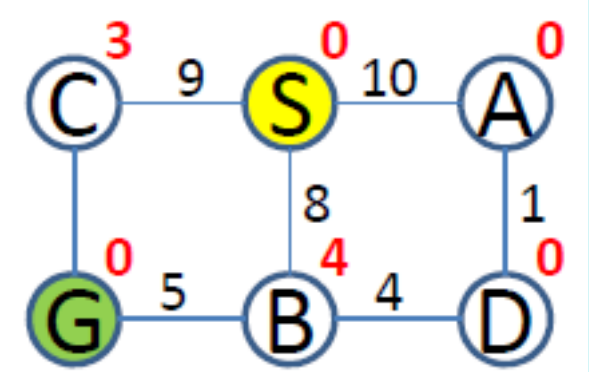


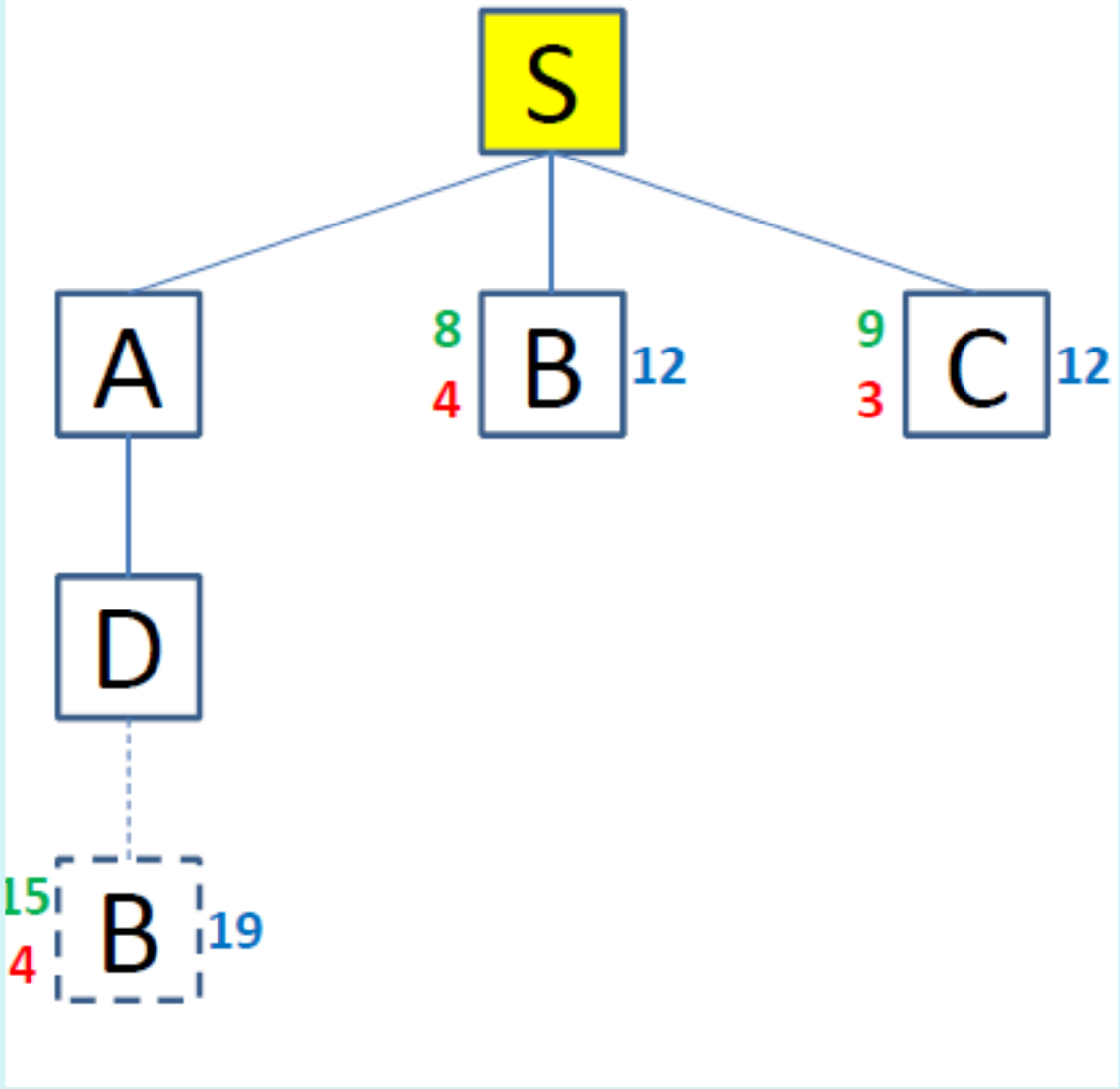


F-bound=12

F-new=∞

IDA* SEARCH

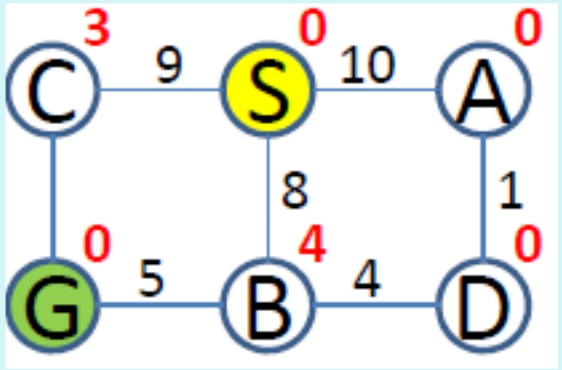


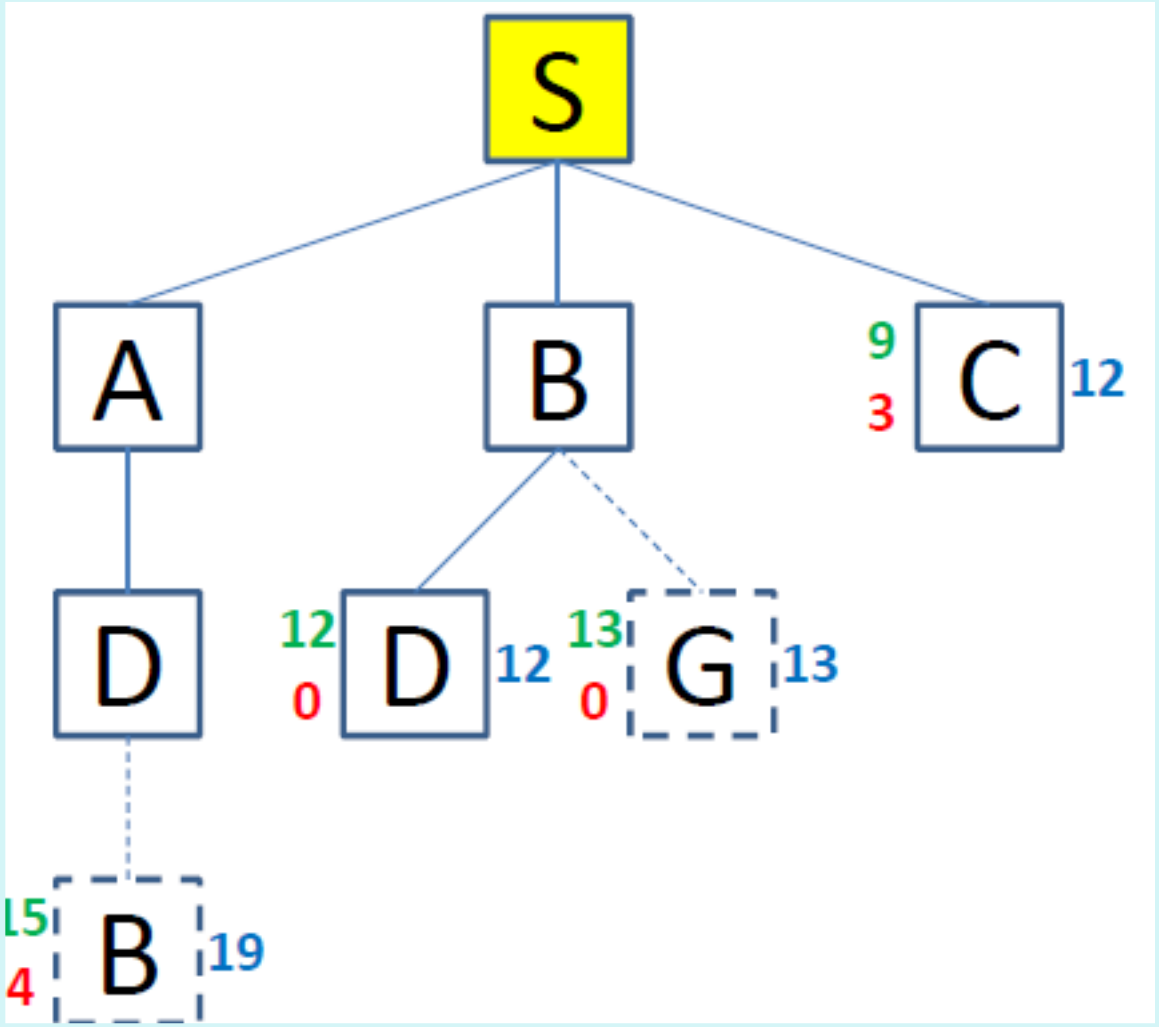


F-bound=12

F-new=19

IDA* SEARCH

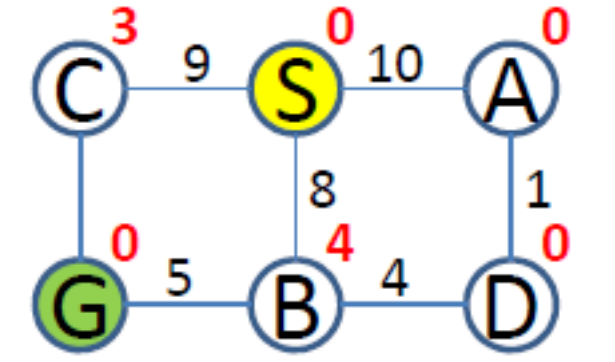


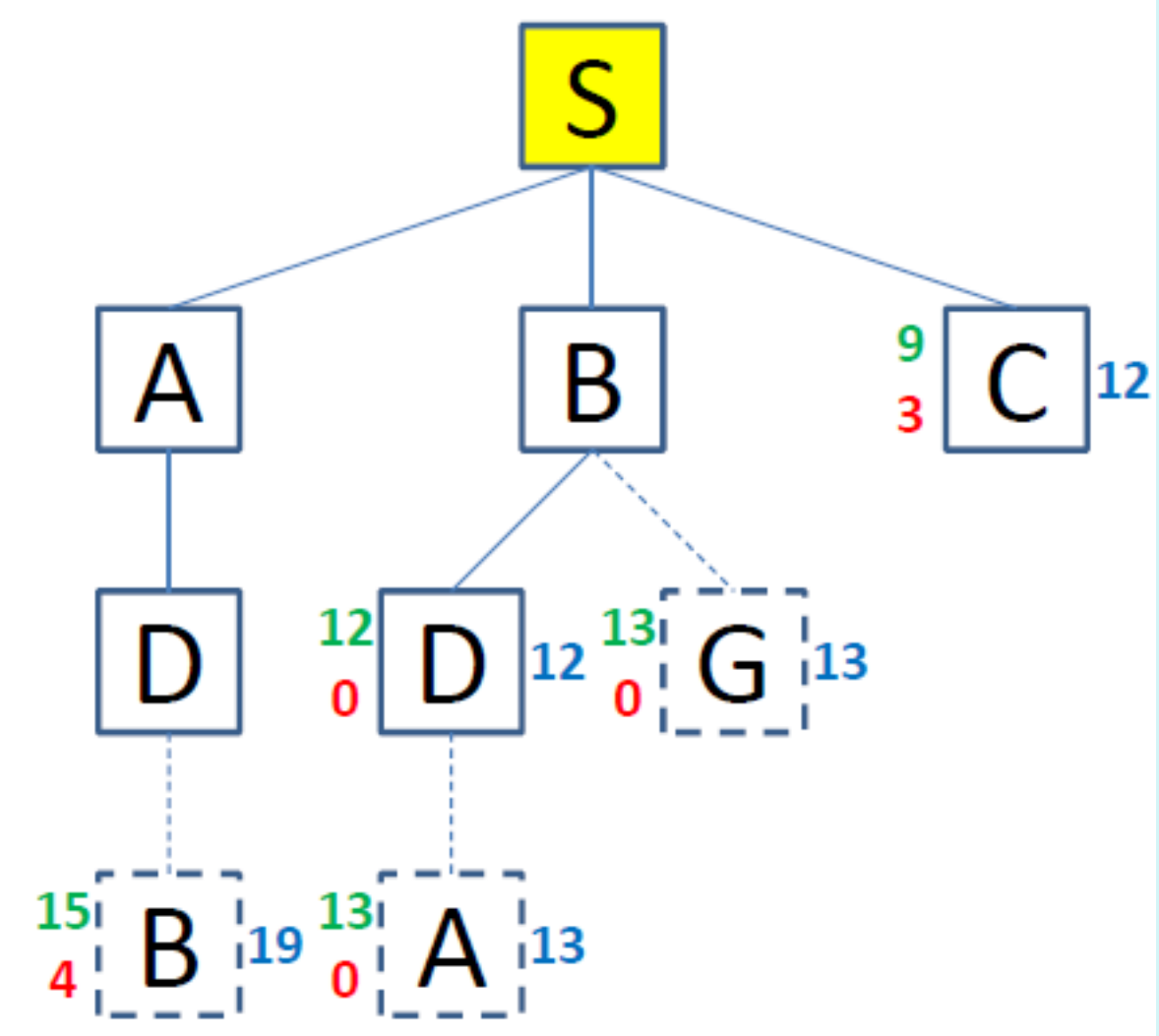


F-bound=12

F-new=13

IDA* SEARCH

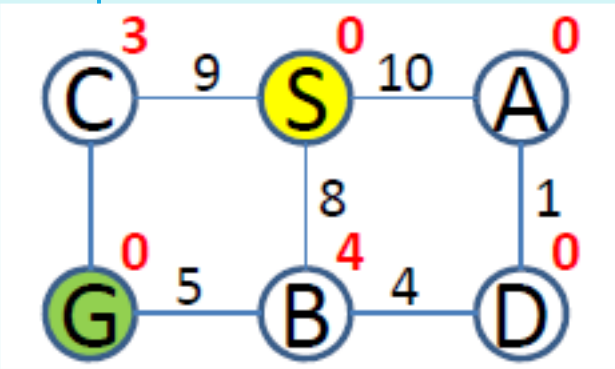


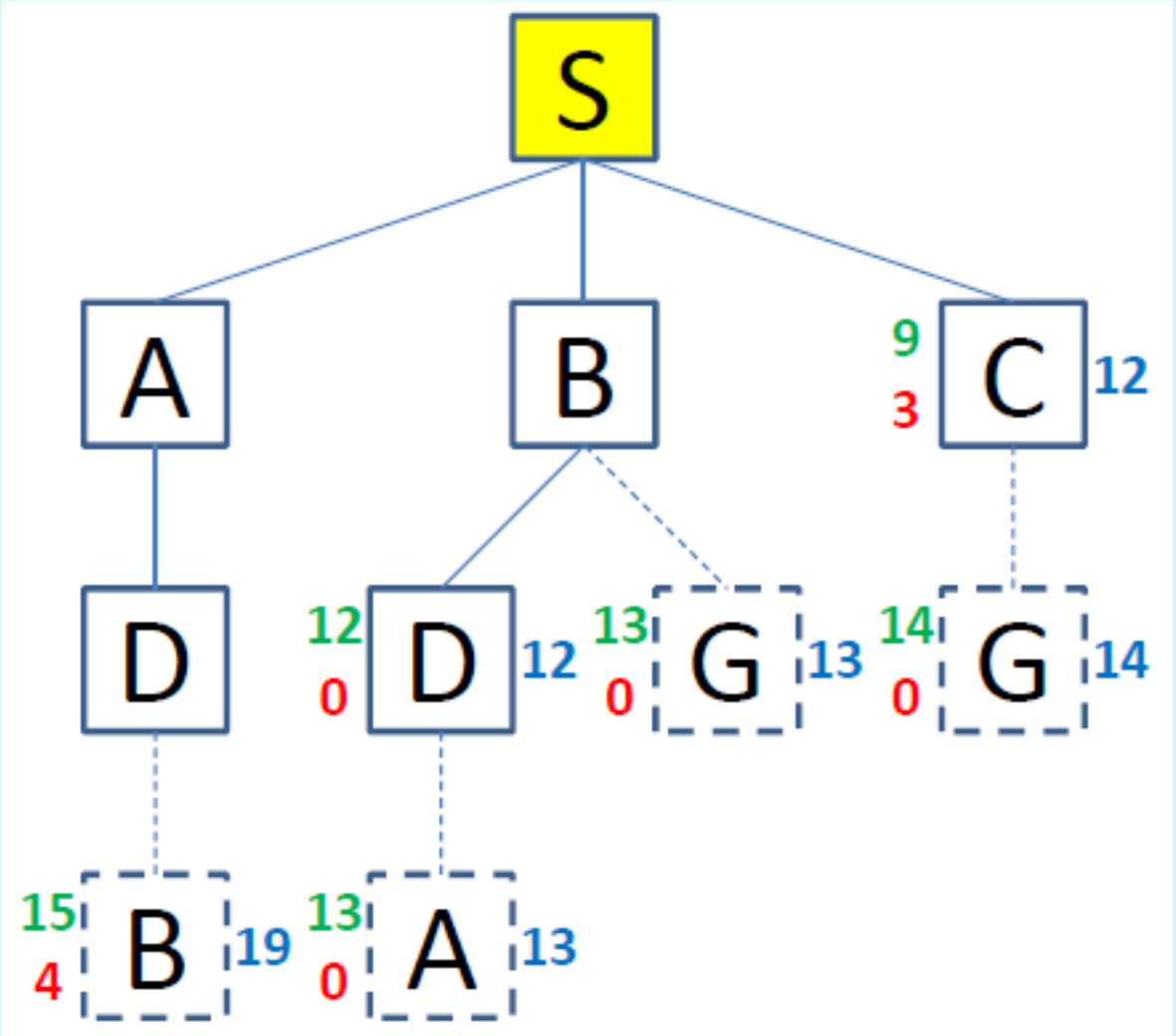


F-bound=12

F-new=13

IDA* SEARCH

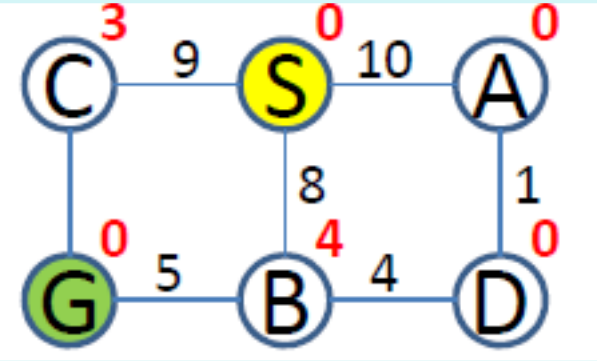




F-bound=12

F-new=13

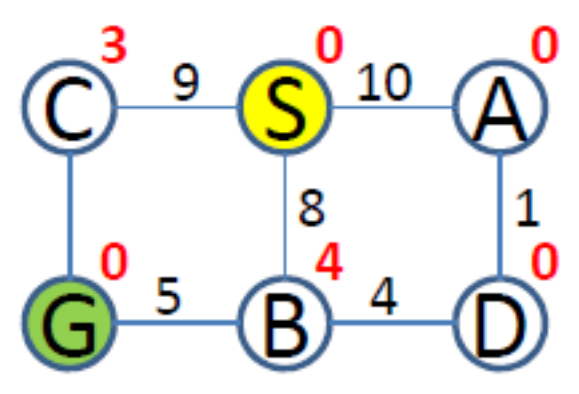
IDA* SEARCH

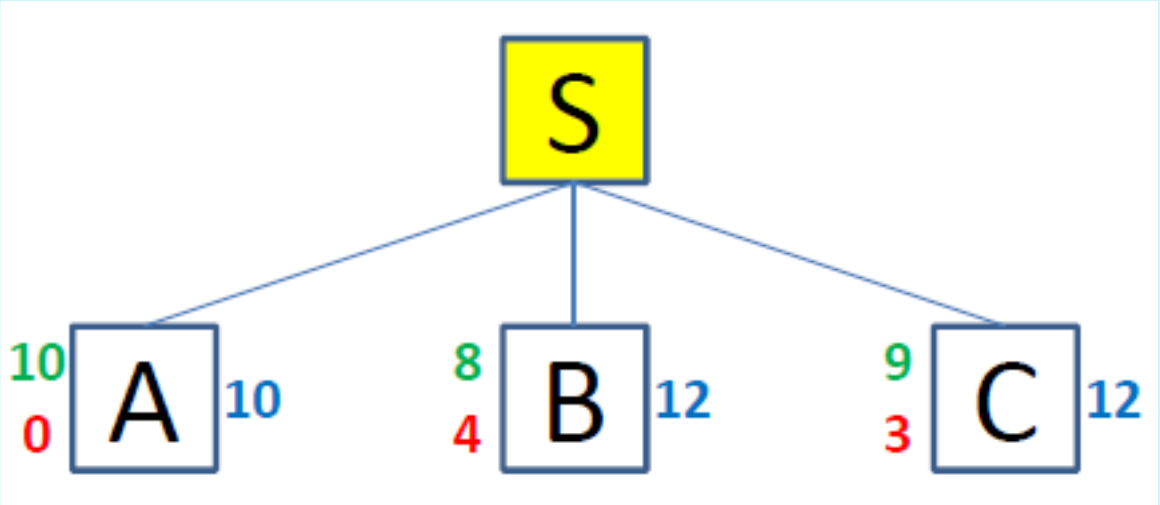


F-bound=13

F-new=∞

IDA* SEARCH

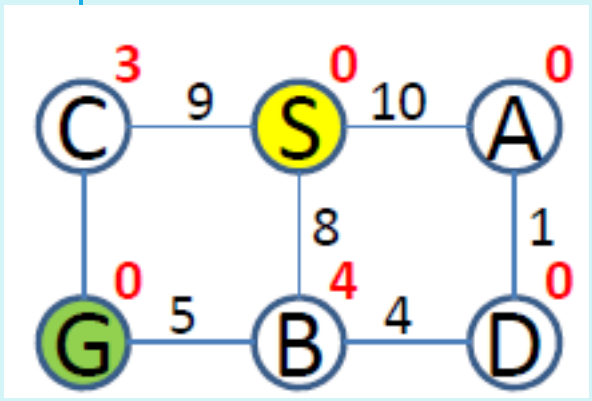


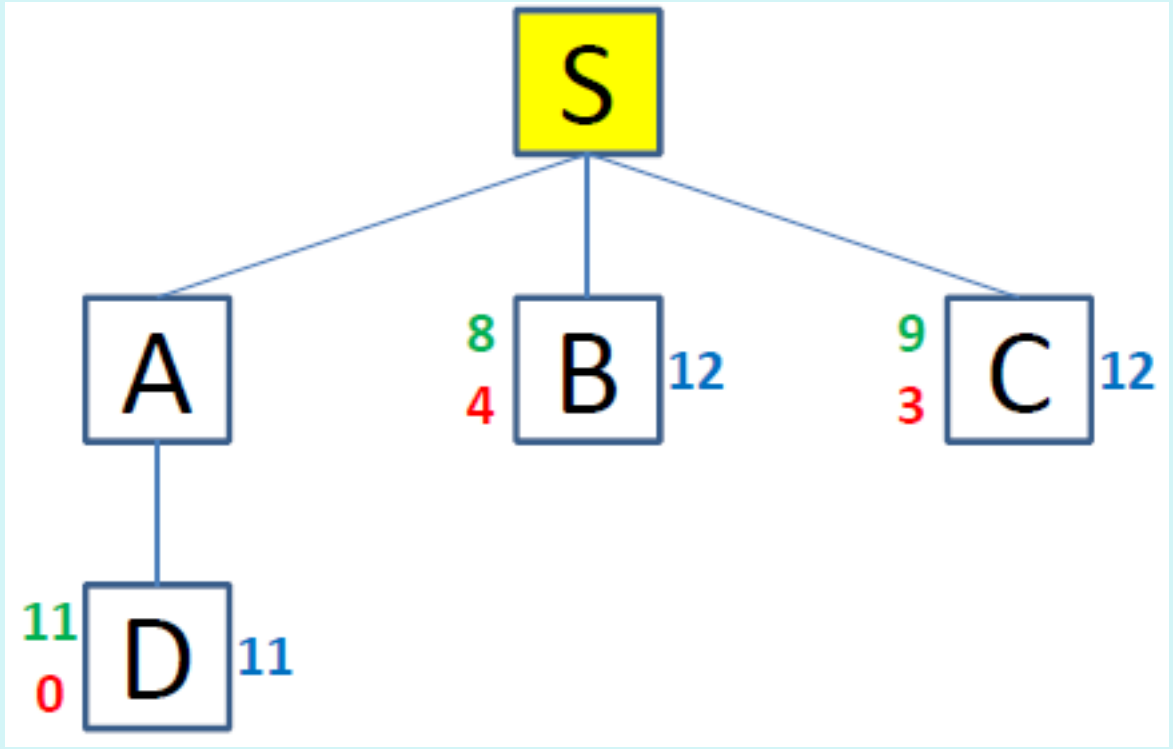


F-bound=13

F-new=∞

IDA* SEARCH

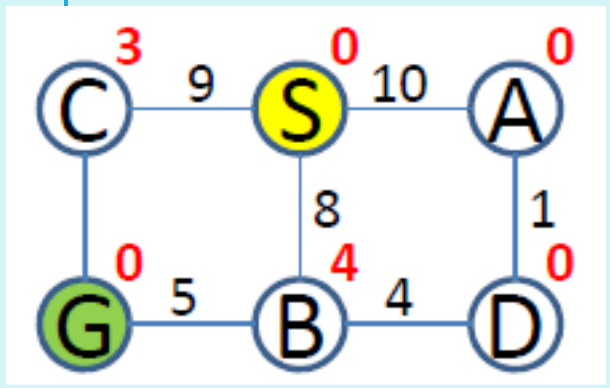


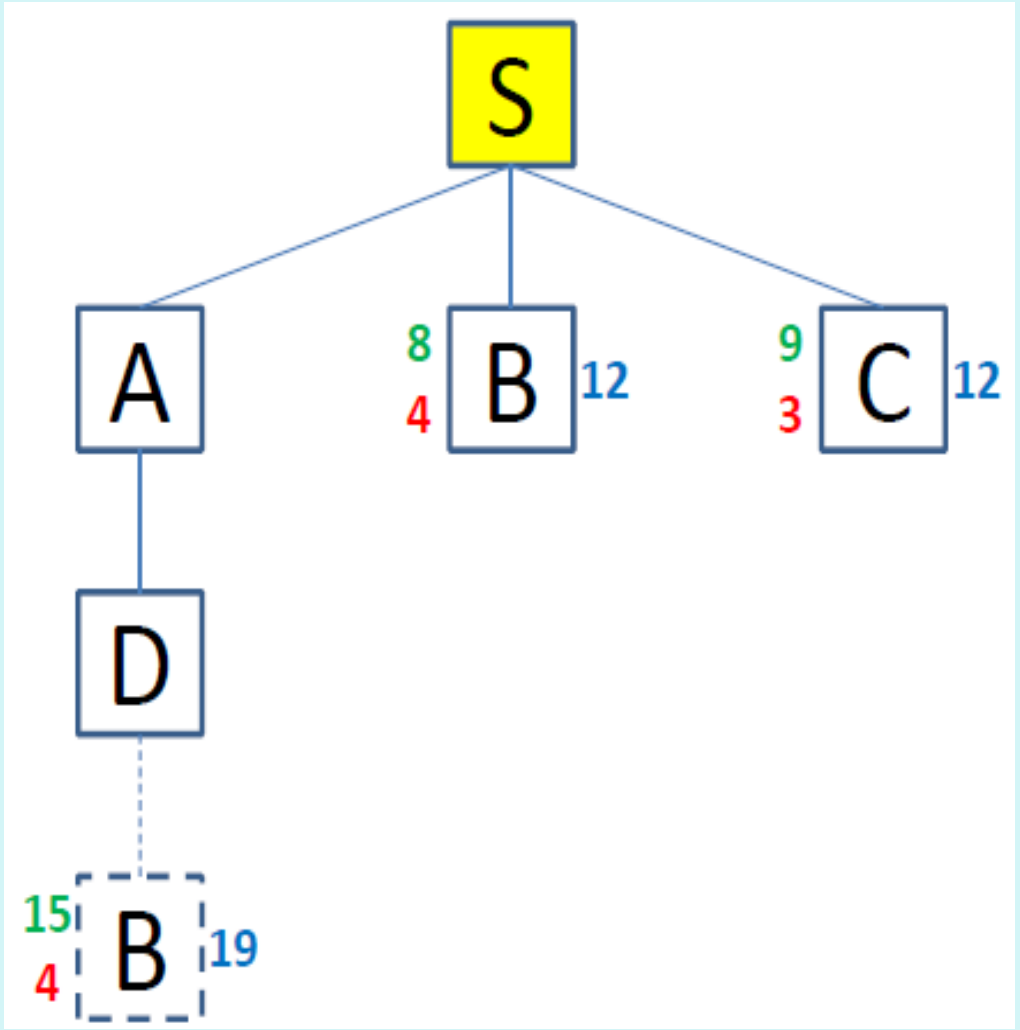


F-bound=13

F-new=∞

IDA* SEARCH

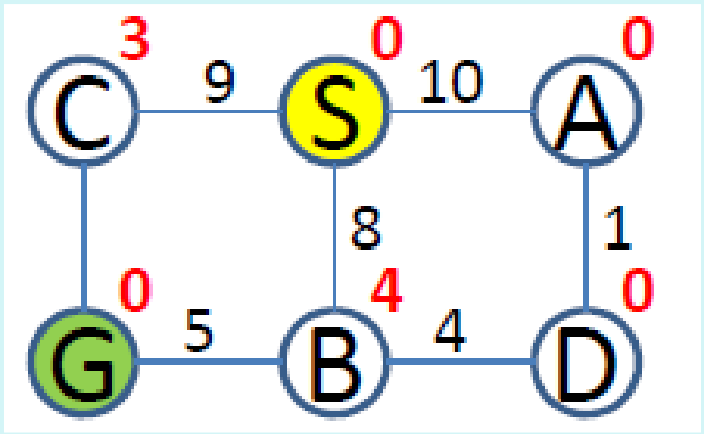


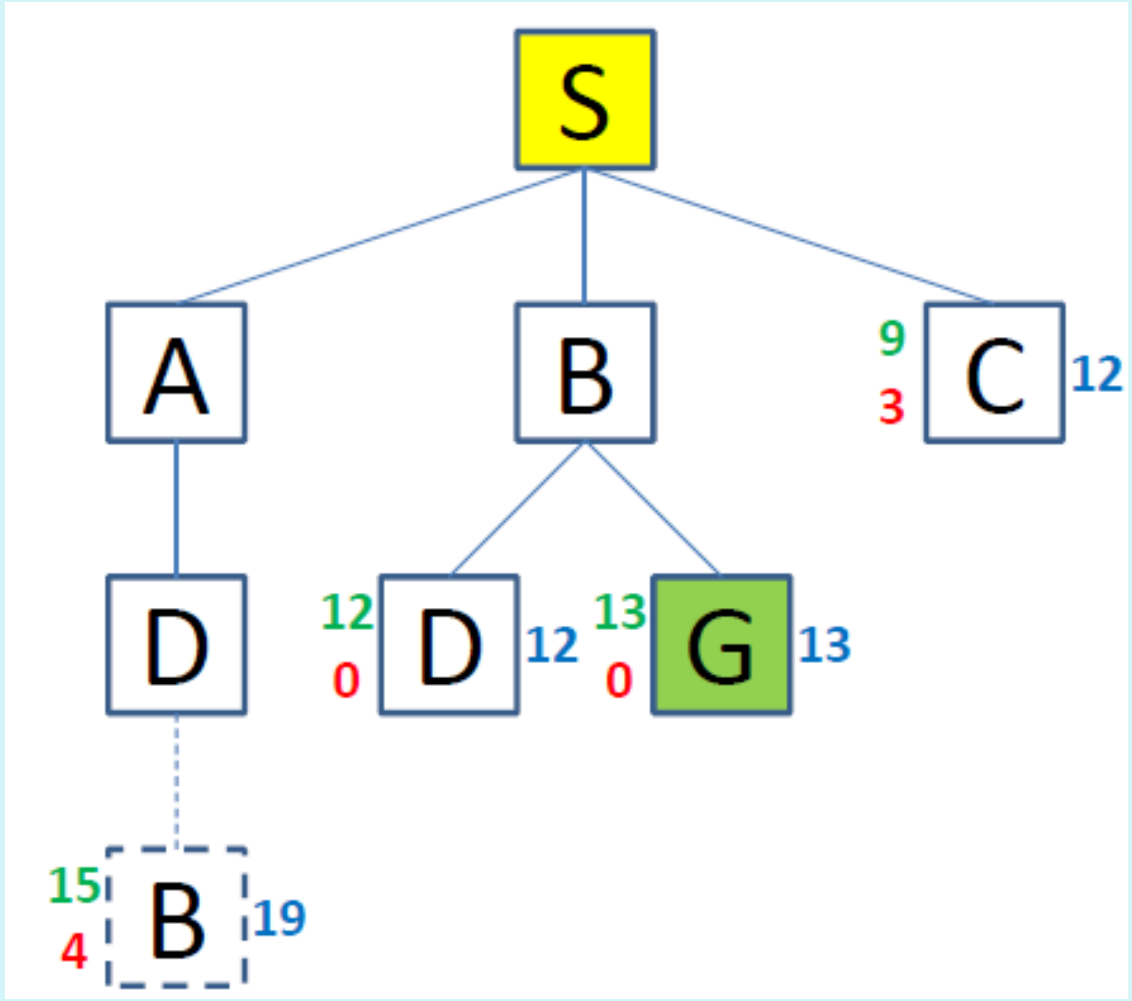


F-bound=13

F-new=19

IDA* SEARCH





F-bound=13

F-new=19

Goal node found, search stops here

IDA* SEARCH ALGORITHM

f-bound ←(S)

Algorithm:

– WHILE (goal is not reached) DO

- f-bound ← f-limited_search(f-bound)
  - – Perform f-limited search with f-bound

F-LIMITED SEARCH ALGORITHM

Input:

- – QUEUE=Path only containing root
- – f-bound←Natural number
- – f-new ← ∞

Algorithm:

- – WHILE (QUEUE not empty && goal not reached) DO
  - Remove first path from QUEUE
  - Create paths to children
  - Reject paths with loops
  - Add paths with f(path) ≤ f-bound to front of QUEUE (depth-first)
  - f-new ← minimum({f-new} U {f(P) | P is rejected path})
- – IF goal reached THEN success ELSE report f-new

ADVANTAGES OF IDA* SEARCH

It will always find the optimal solution provided that it exists and that if a heuristic is supplied it must be admissible.

Heuristic is not necessary, it is used to speed up the process.

Various heuristics can be integrated to the algorithm without changing the basic code.

The cost of each move can be tweaked into the algorithms as easily as the heuristic

Uses a lot less memory which increases linearly as it doesn't store and forgets after it reaches a certain depth and start over again.

LIMITATIONS OF IDA* SEARCH

It reduces storage requirement but increases the time.

No sense of direction, many nodes are explored multiple times.

It takes exponential amount of time if nodes are well connected.

IDA* is better suited for the less connected graphs and not others.

When we take an admissible heuristic function, it does not mean that we have the best possible heuristic function.

CHAPTER 2 COMPLETED