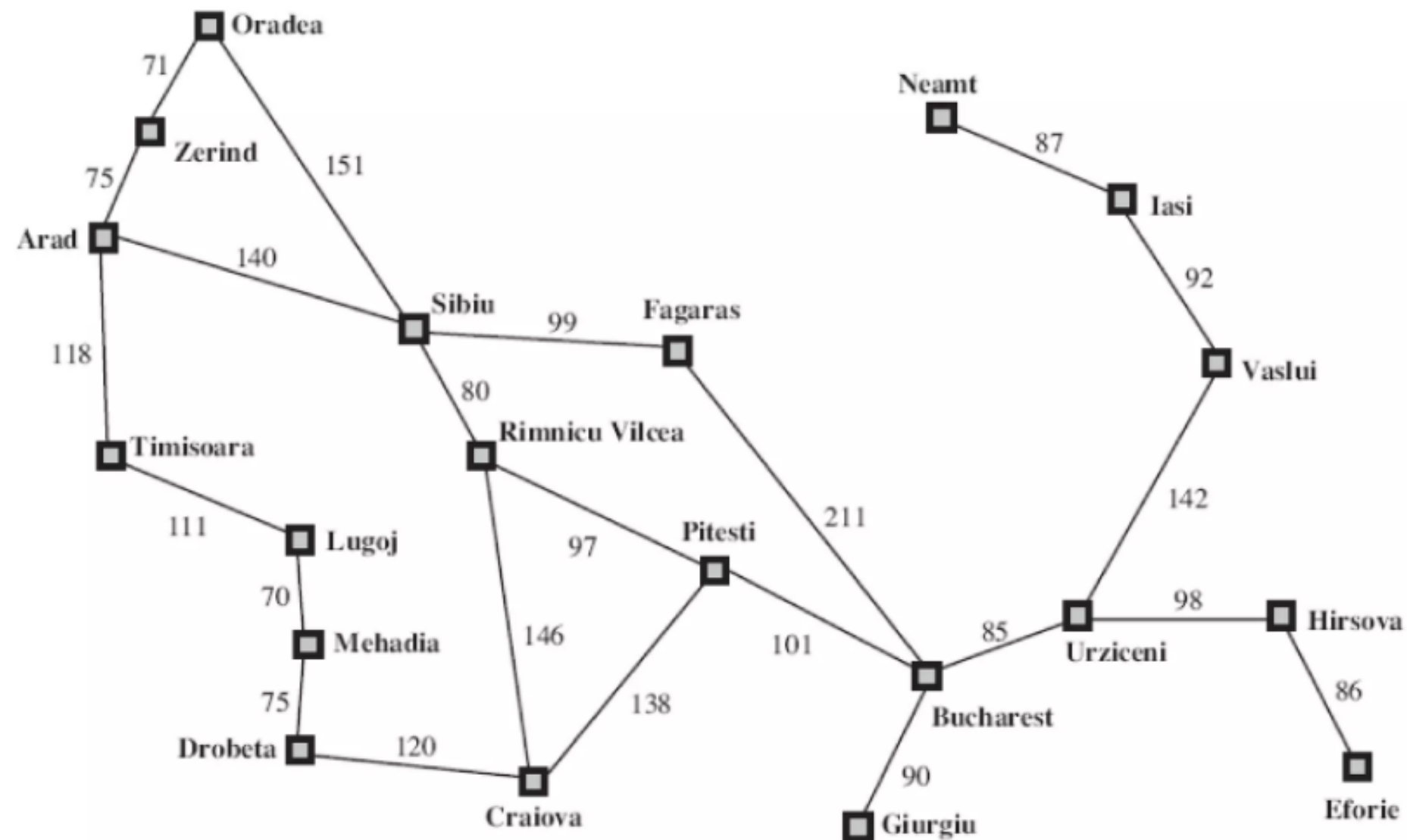


MEMORY BOUNDED HEURISTIC SEARCH

Recursive Best First Search

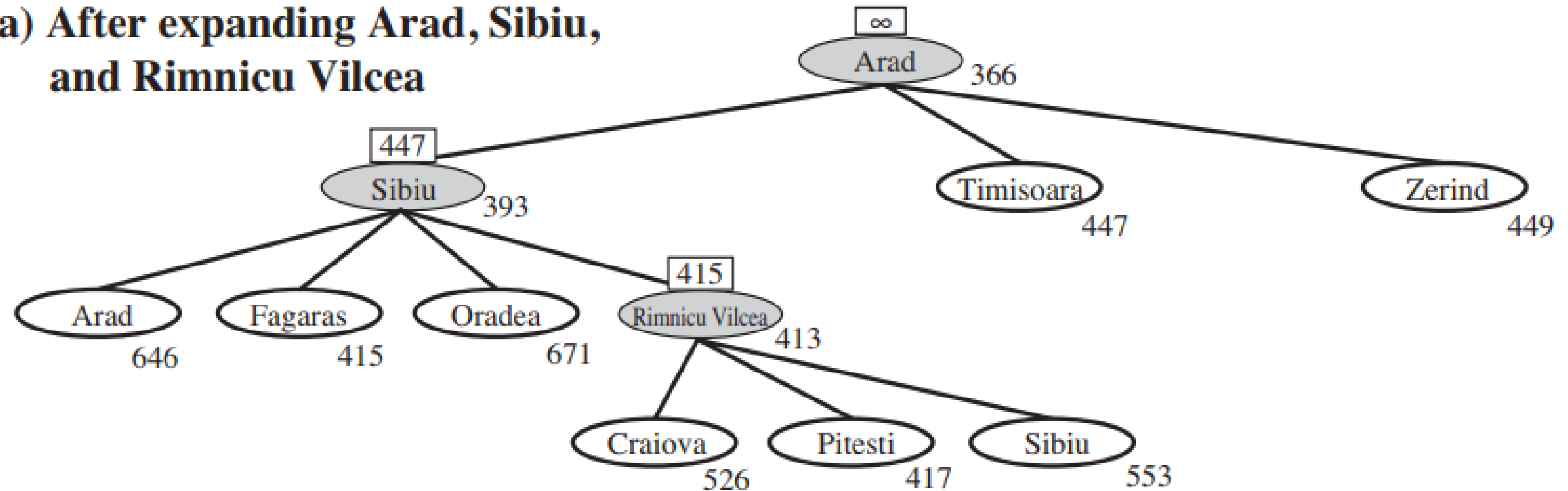


H-values

Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

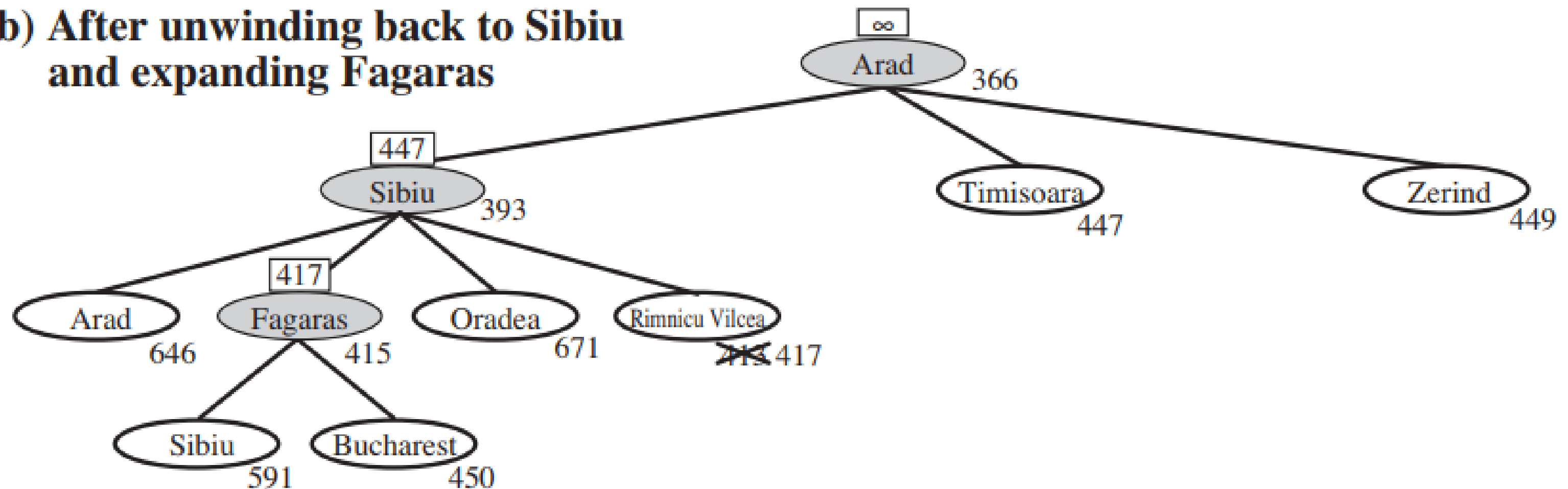
Recursive Best First Search

(a) After expanding Arad, Sibiu, and Rimnicu Vilcea



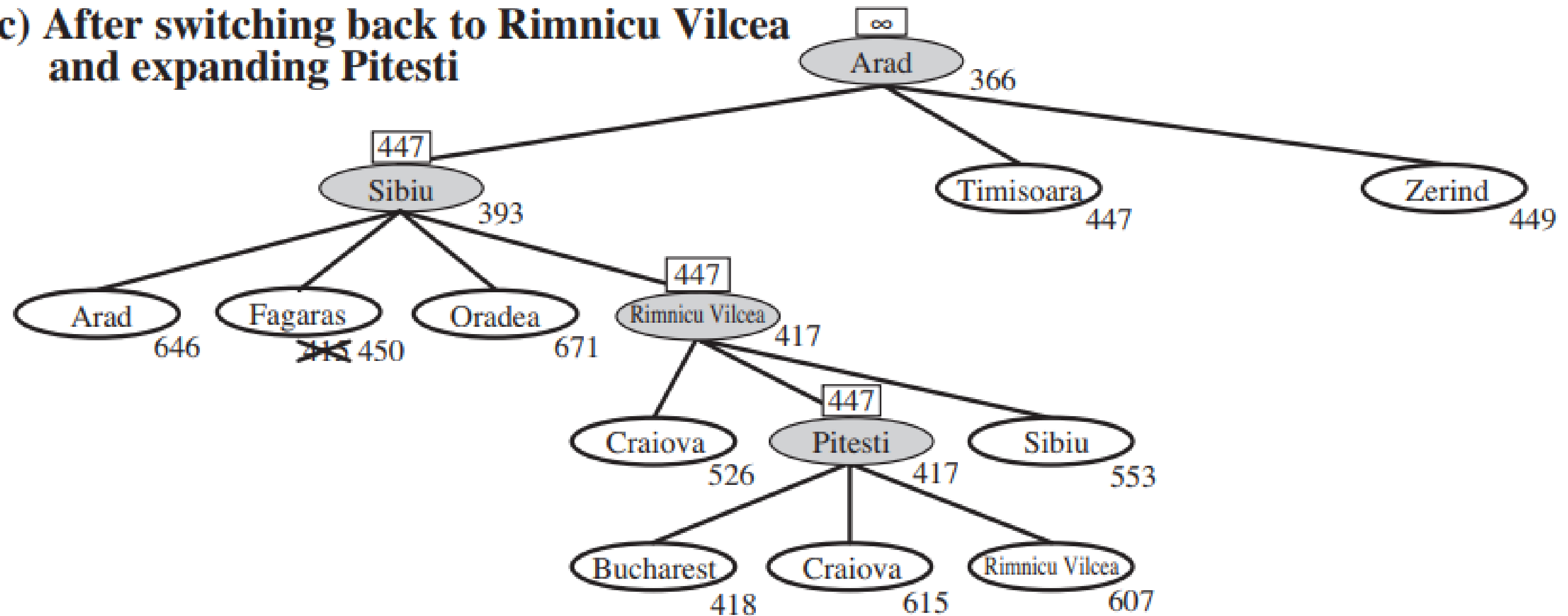
Recursive Best First Search

(b) After unwinding back to Sibiu and expanding Fagaras

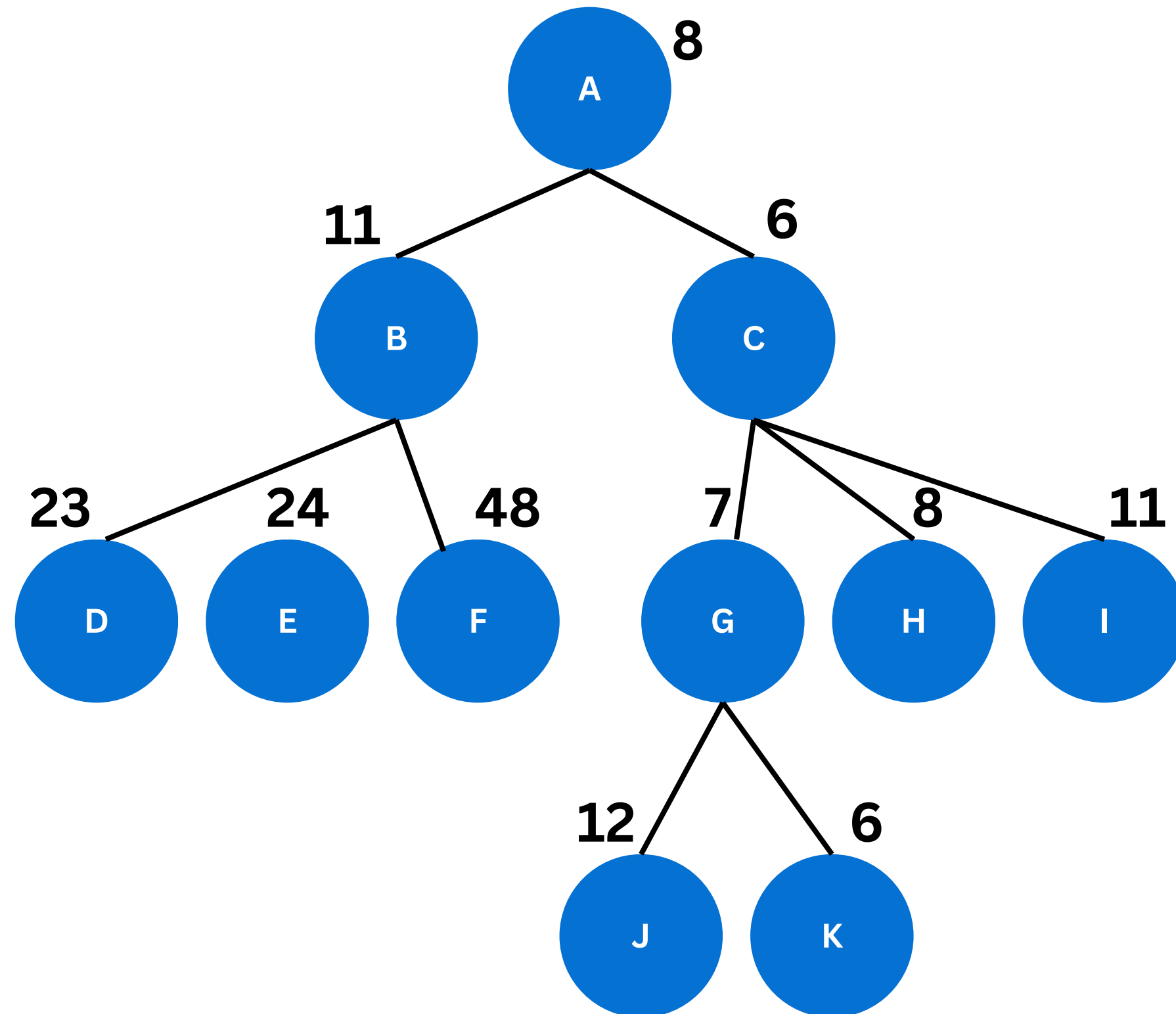


Recursive Best First Search

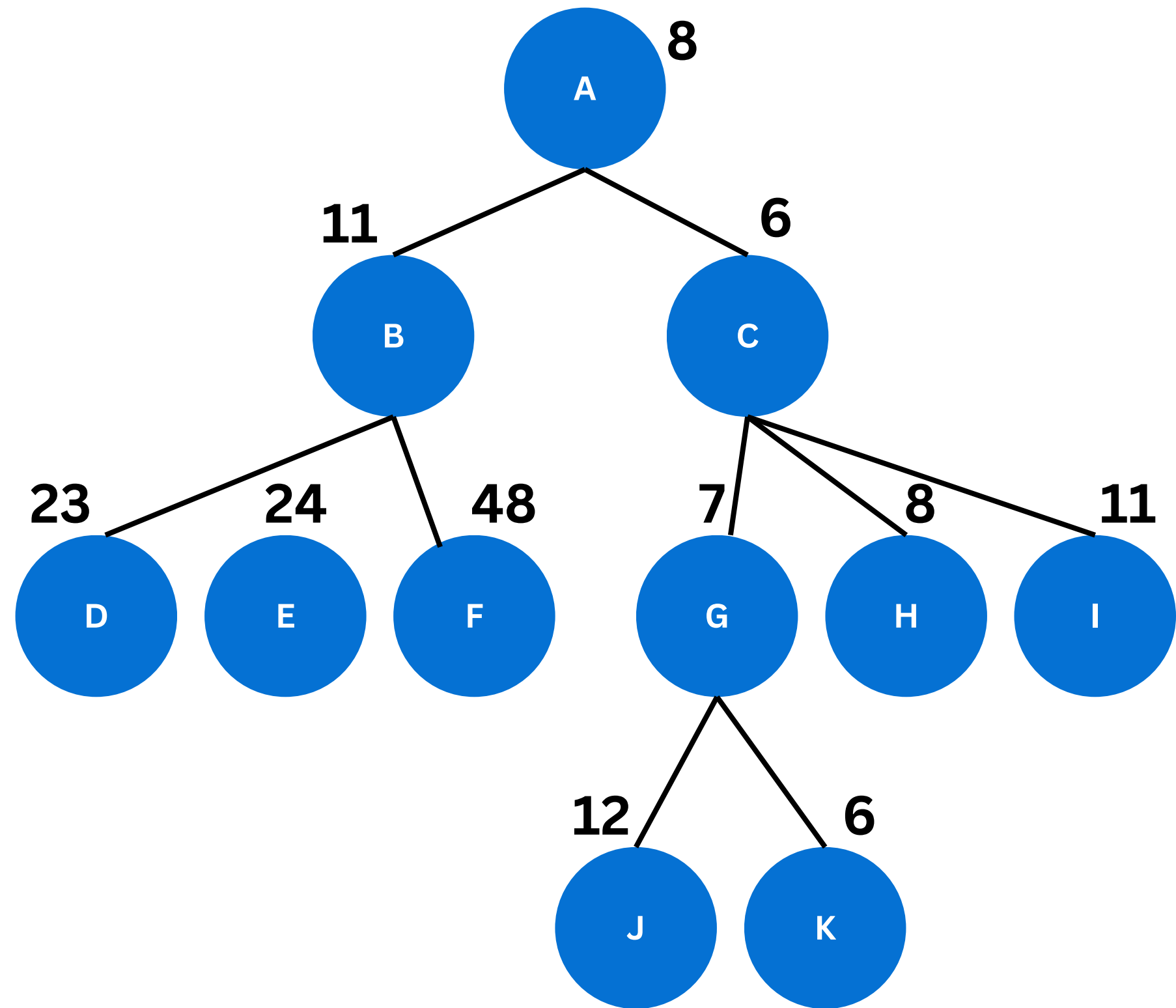
(c) After switching back to Rimnicu Vilcea and expanding Pitesti



ITERATIVE DEEEPENING A*



ITERATIVE DEEPENING A*



Threshold = 6

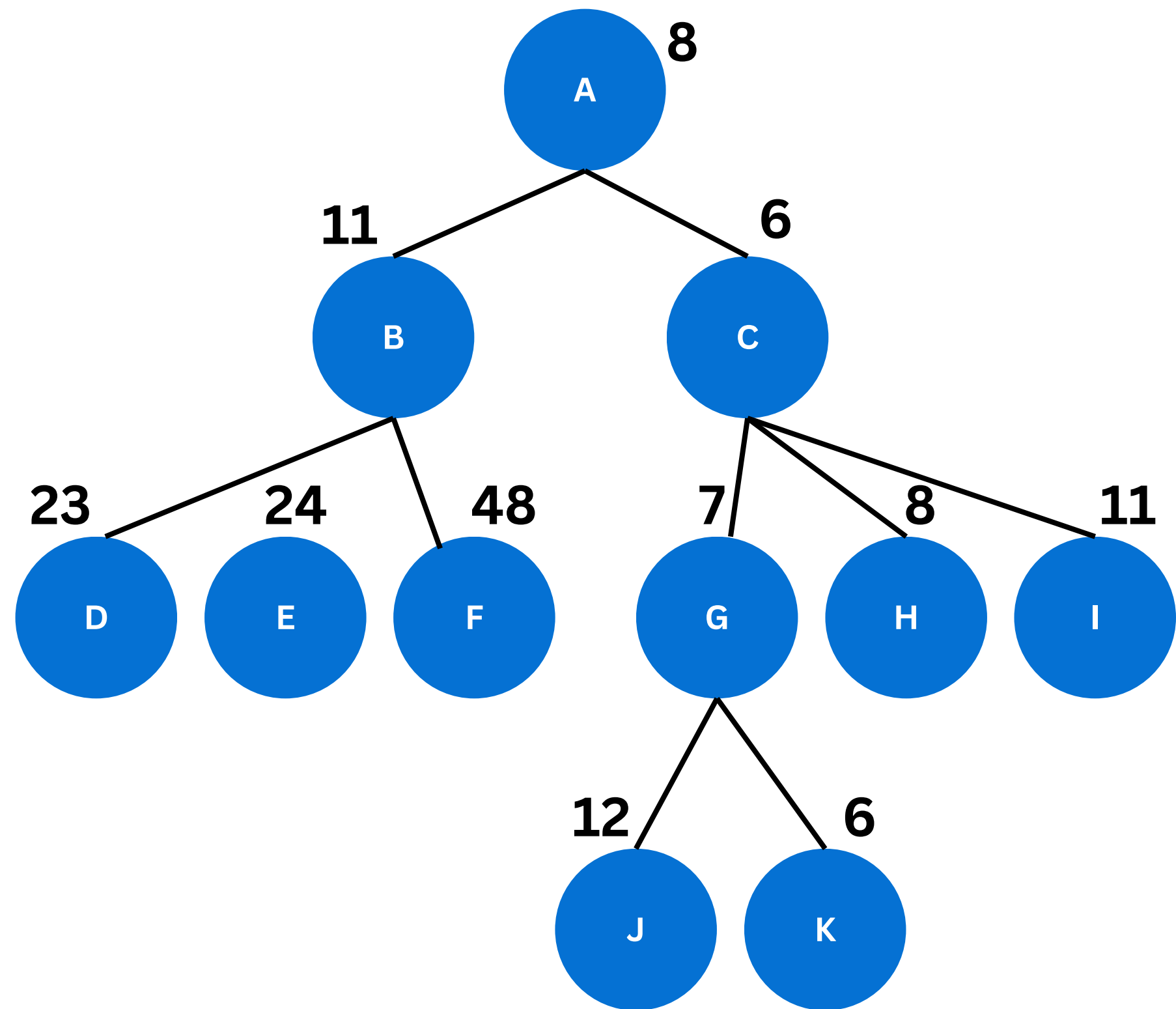
A's f-score is

$8 > \text{threshold}$

so change the

threshold

ITERATIVE DEEPENING A*



Threshold = 7

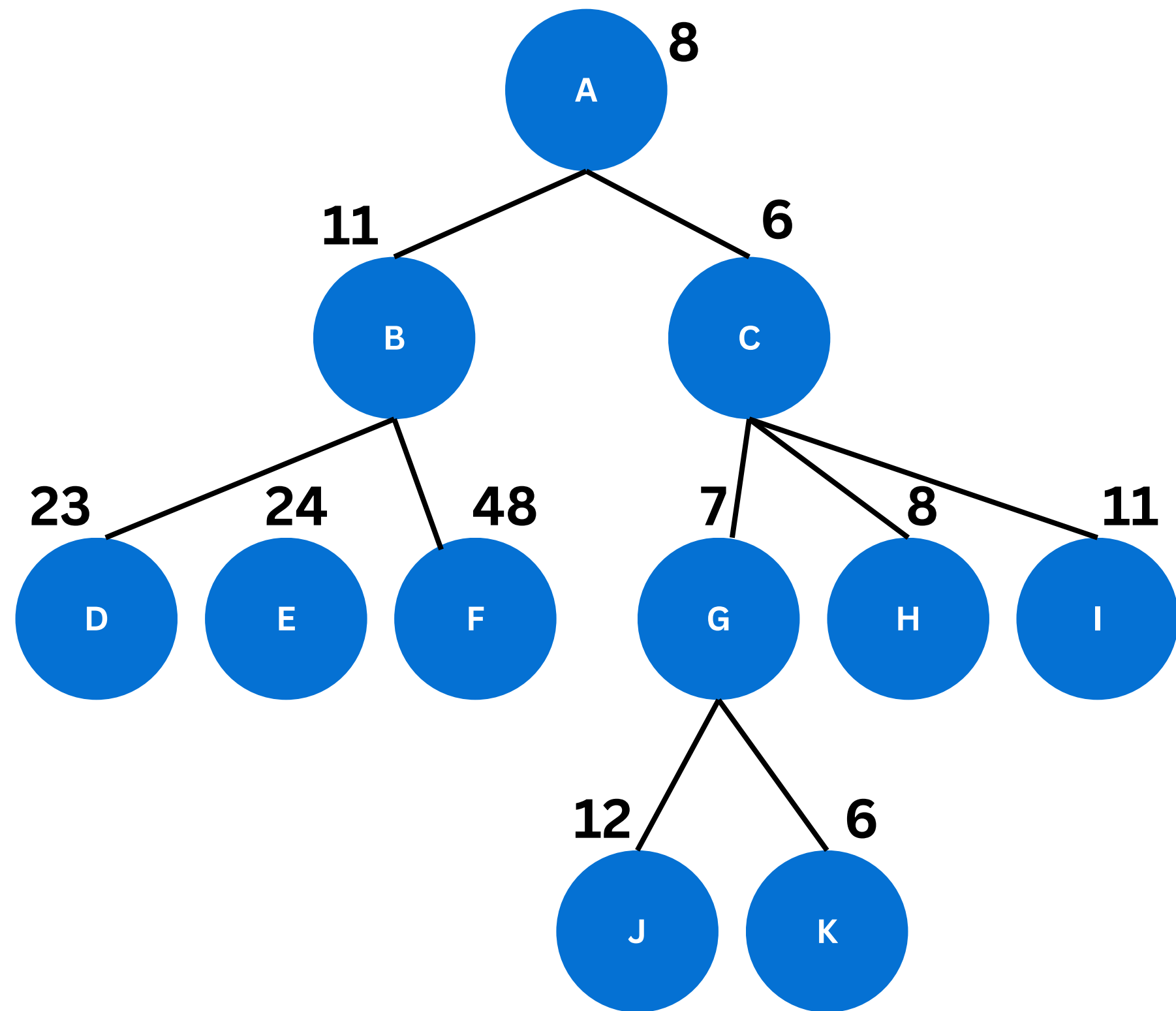
A's f-score is

$8 > \text{threshold}$

**so change the
threshold**

still,

ITERATIVE DEEPENING A*

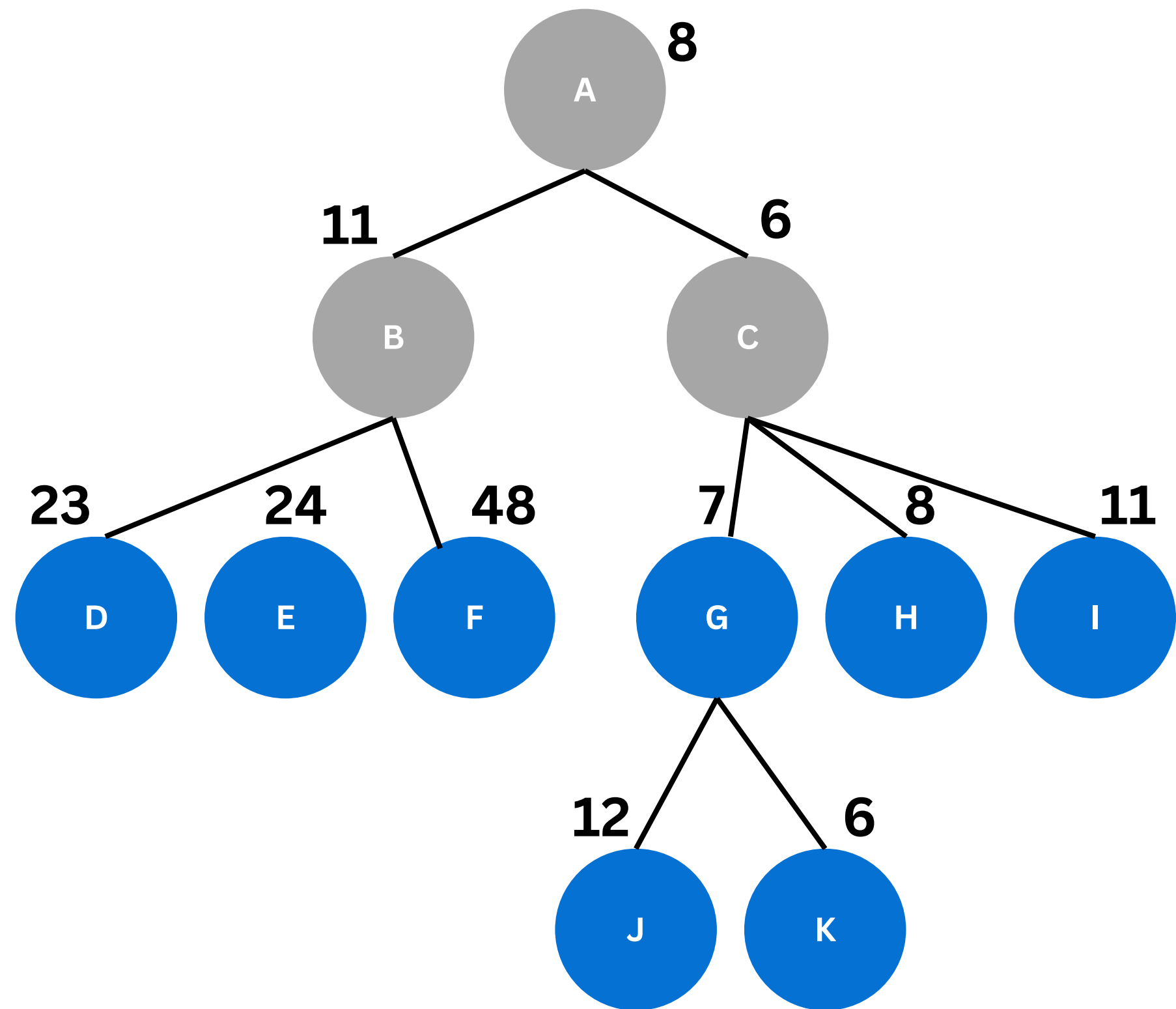


Threshold = 8

**A's f-score is
8 ≤ threshold**

equal is fine, lets expand A

ITERATIVE DEEPENING A*

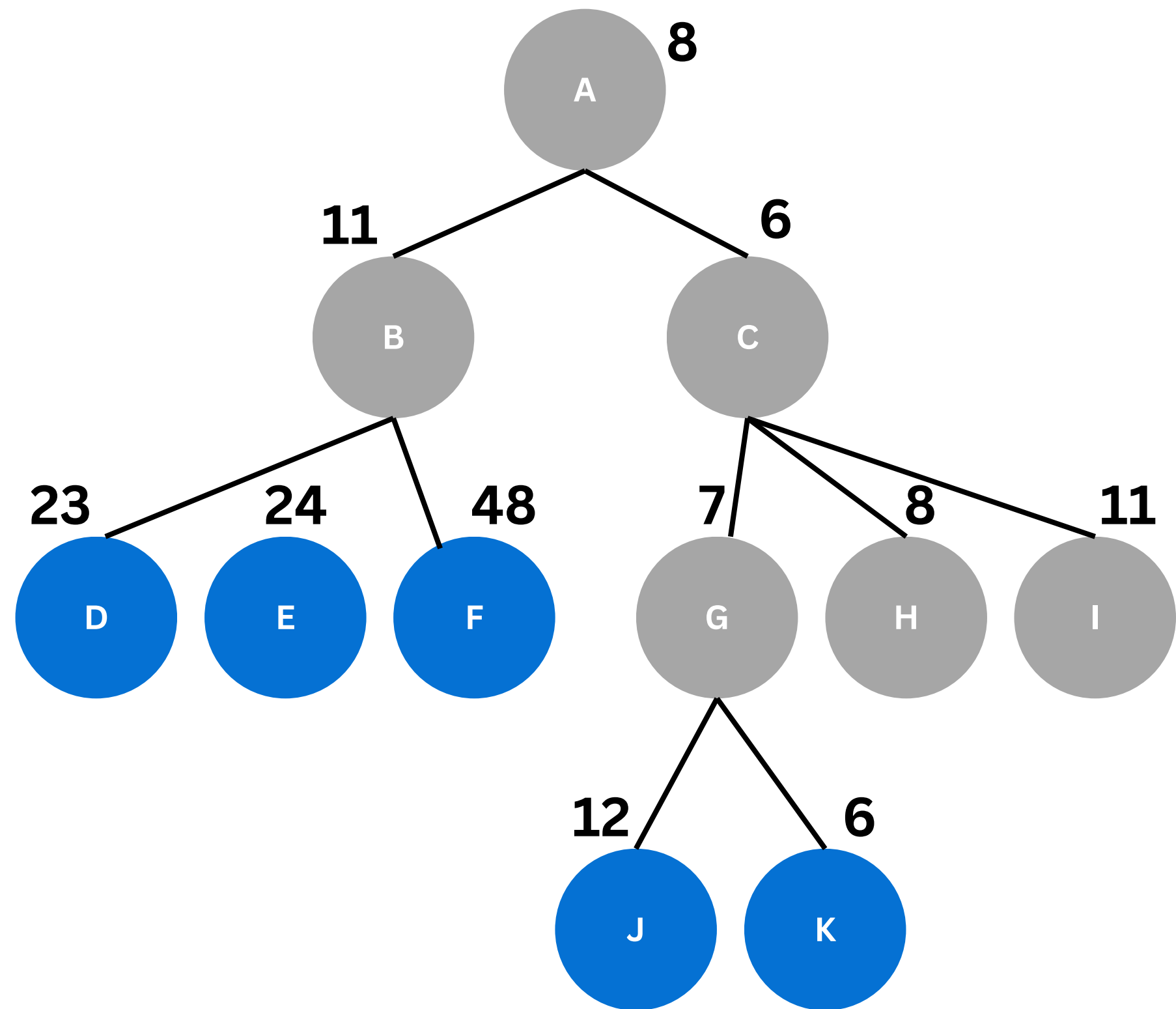


Threshold = 8

**c's f-score is
 $6 \leq \text{threshold}$**

expand C

ITERATIVE DEEPENING A*

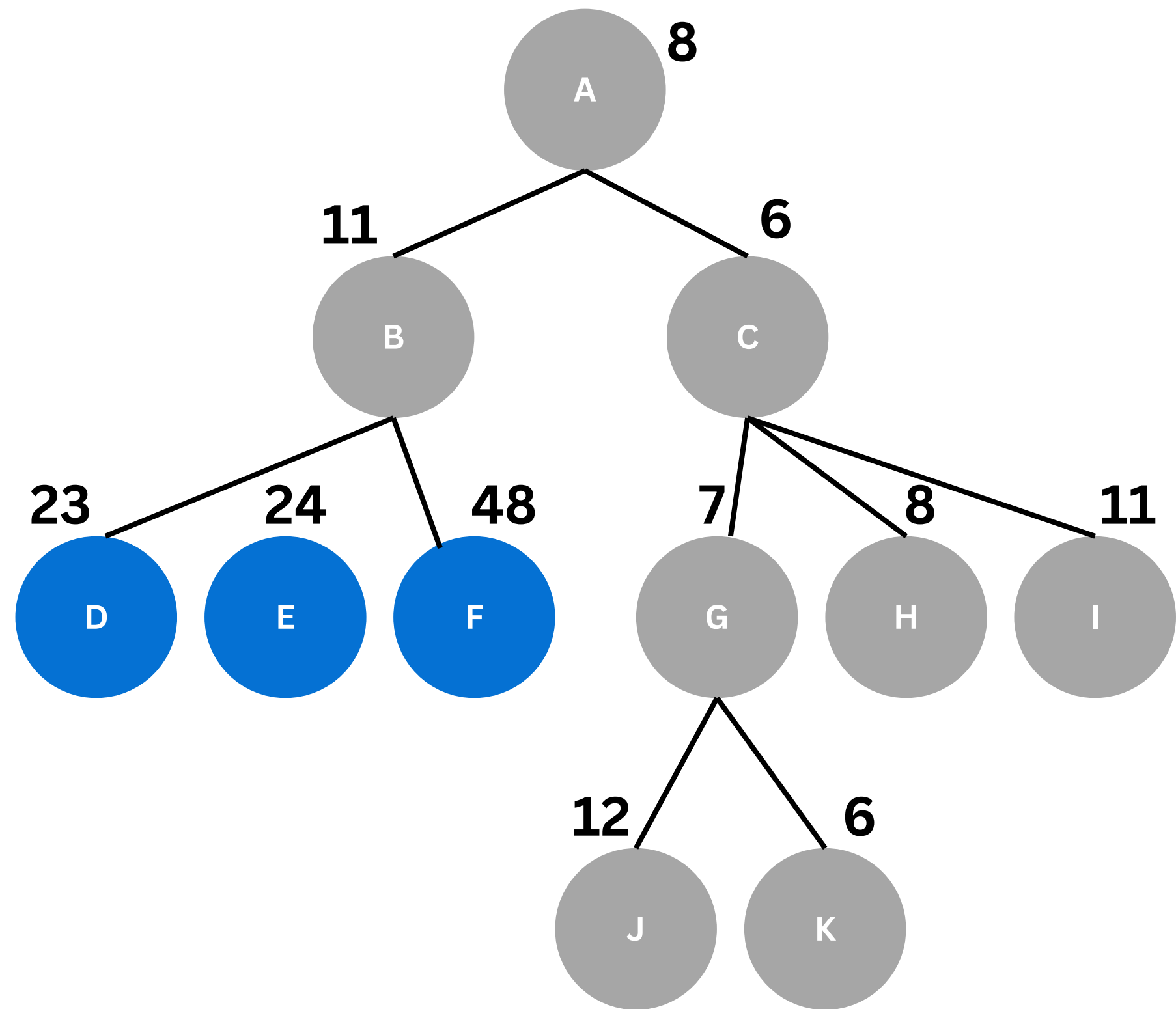


Threshold = 8

**c's f-score is
 $6 \leq \text{threshold}$**

**now the small
f(n) value G,
expand that**

ITERATIVE DEEPENING A*



Threshold = 8

**c's f-score is
 $6 \leq \text{threshold}$**

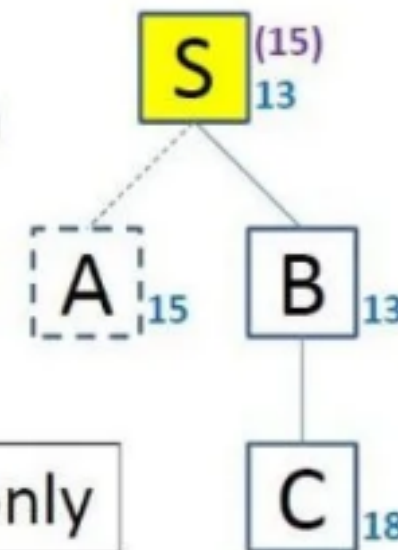
**lessgooooo
we reached K
from A**

(AK's path, haha)

SIMPLIFIED MEMORY BOUNDED A*

SMA* Algorithm

- Optimizes A* to work within reduced memory
- **Key Idea:**
 - IF memory **full** for **extra node (C)**
 - **Remove** highest f-value leaf (**A**)
 - **Remember** best-forgotten child in each parent node (**15 in S**)



E.g. Memory of 3 nodes only

Image 1: Idea of how SMA* works

SIMPLIFIED MEMORY BOUNDED A*

SMA* Algorithm

- **Generate Children 1 by 1**
 - **Expanding:** add 1 child at the time to QUEUE
 - Avoids **memory overflow**
 - **Allows monitoring** if nodes need deletion

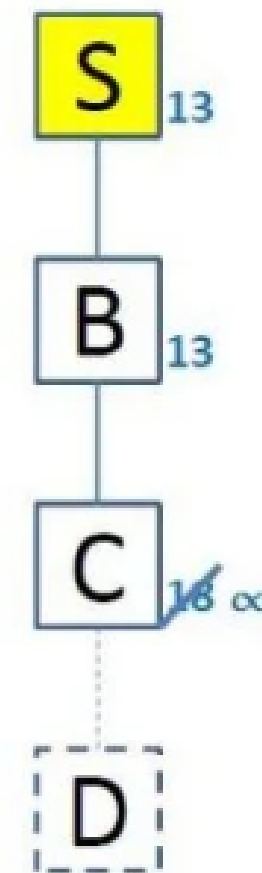


Image 2: Generating Children in SMA* Algorithm

SIMPLIFIED MEMORY BOUNDED A*

SMA* Algorithm

- **Too long paths: Give up**
 - Extending path cannot fit in memory
 - give up (C)
 - Set **f-value** node (C) to ∞
 - **Remembers:**
path cannot be found here



E.g. Memory of 3 nodes only

Image 3: Handling Long Paths i.e. Too Many Nodes In The Memory

SIMPLIFIED MEMORY BOUNDED A*

SMA* Algorithm

- **Adjust f-values**
 - IF all children M_i of node N have been explored
 - AND $\forall i: f(S...M_i) > f(S...N)$
 - THEN reset (through $N \implies$ through children)
 - $f(S...N) = \min\{f(S...M_i) \mid M_i \text{ child of } N\}$

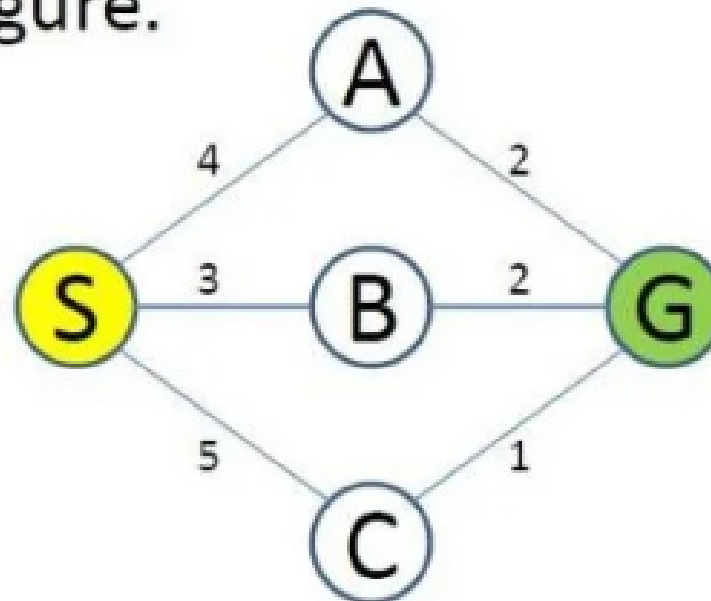


Image 4: Adjusting The f Value

SIMPLIFIED MEMORY BOUNDED A*

SMA* by Example

- Perform SMA* (memory: 3 nodes) on the following figure.

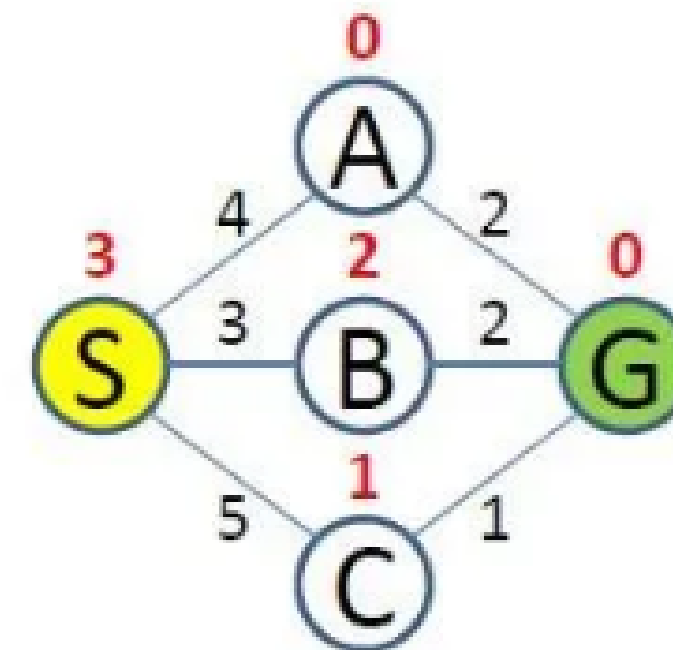
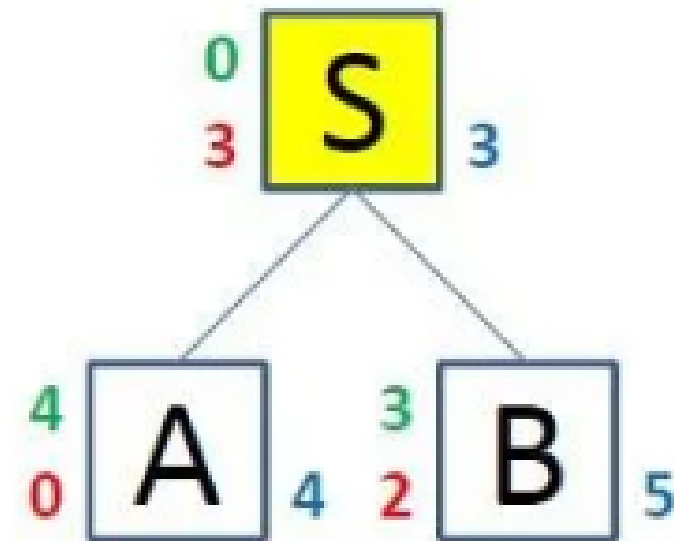


	S	A	B	C	G
heuristic	3	0	2	1	0

Image 5: The Problem of Simplified Memory Bounded A*

SIMPLIFIED MEMORY BOUNDED A*

SMA* by Example

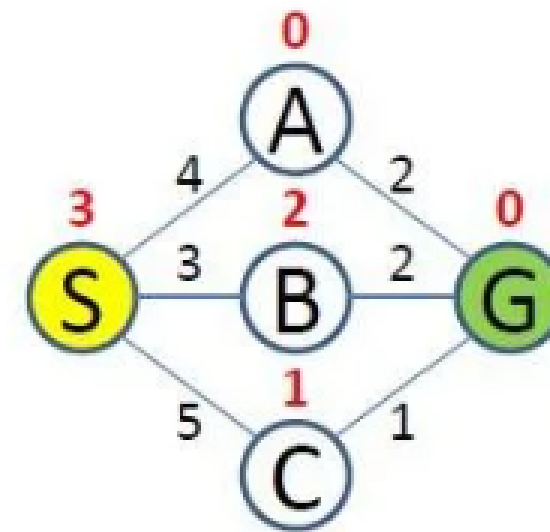
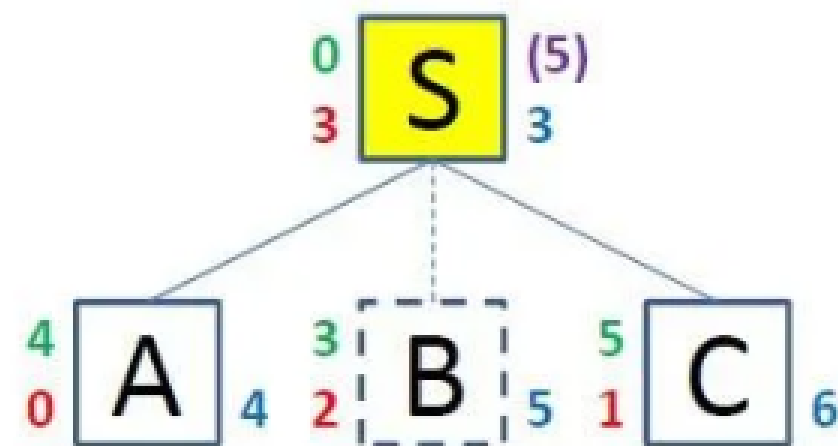


Generate children
(One by one)

Image 6: Evaluating The Children

SIMPLIFIED MEMORY BOUNDED A*

SMA* by Example



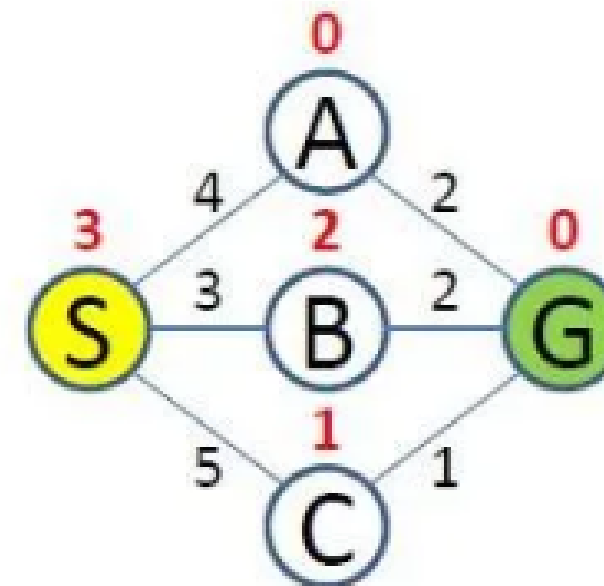
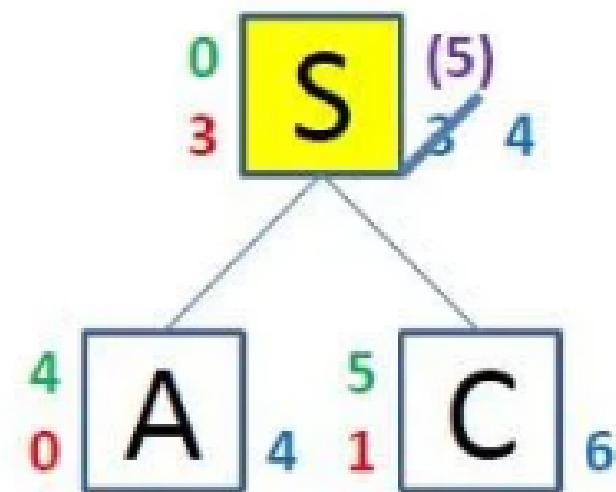
Generate children
(One by one)

Memory full

Image 7: Full Memory While Evaluation

SIMPLIFIED MEMORY BOUNDED A*

SMA* by Example



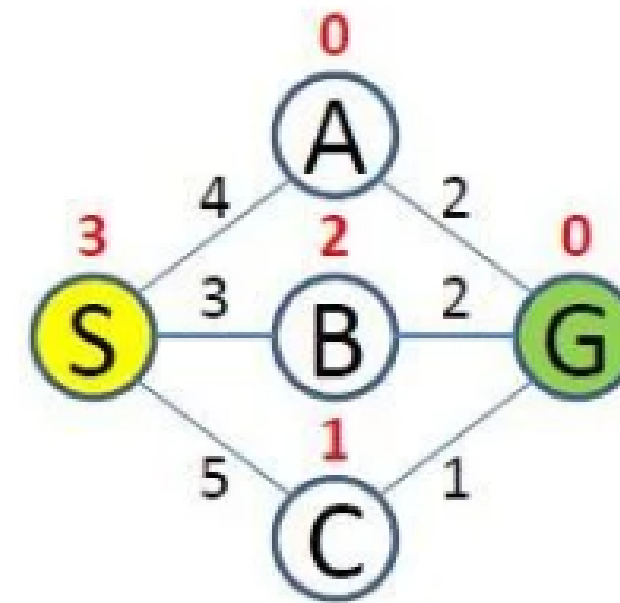
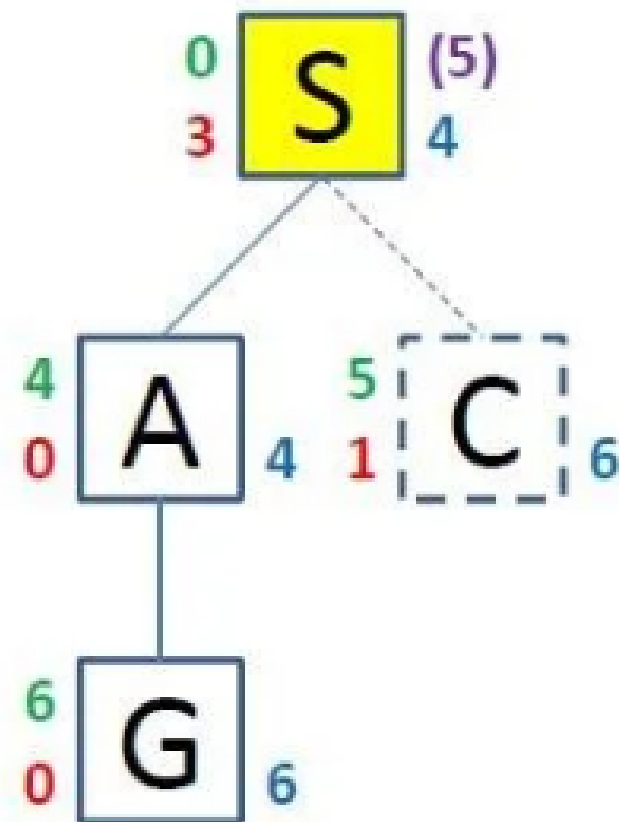
All children are
explored

Adjust f-values

Image 8: Adjusting The f Value

SIMPLIFIED MEMORY BOUNDED A*

SMA* by Example



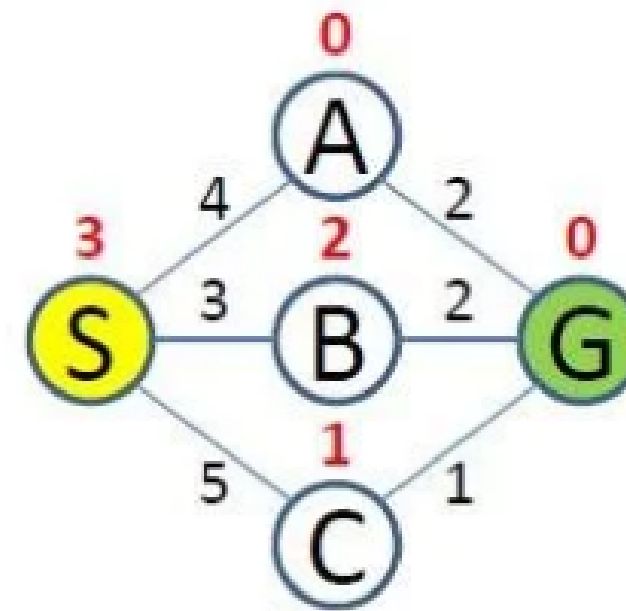
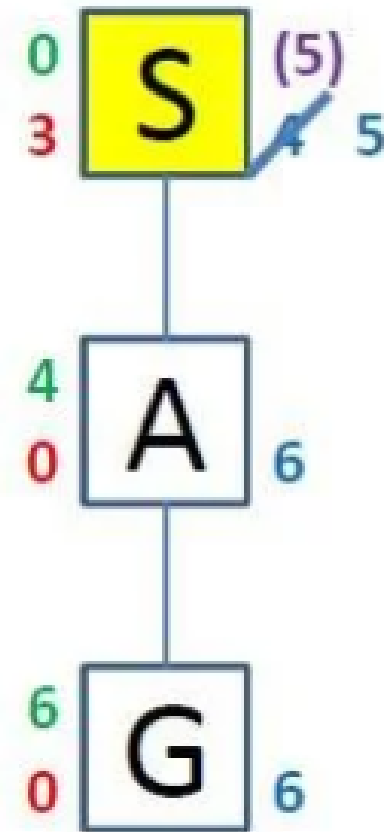
Generate children
(One by one)

Memory full

Image 9: Memory Full Evaluate Through The Lowest f Value Child

SIMPLIFIED MEMORY BOUNDED A*

SMA* by Example



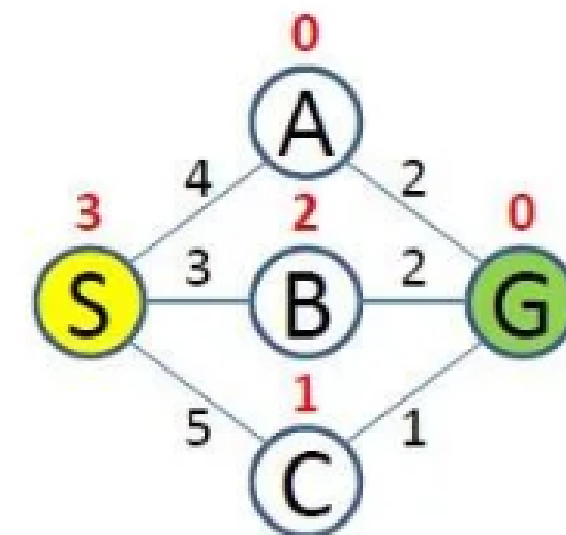
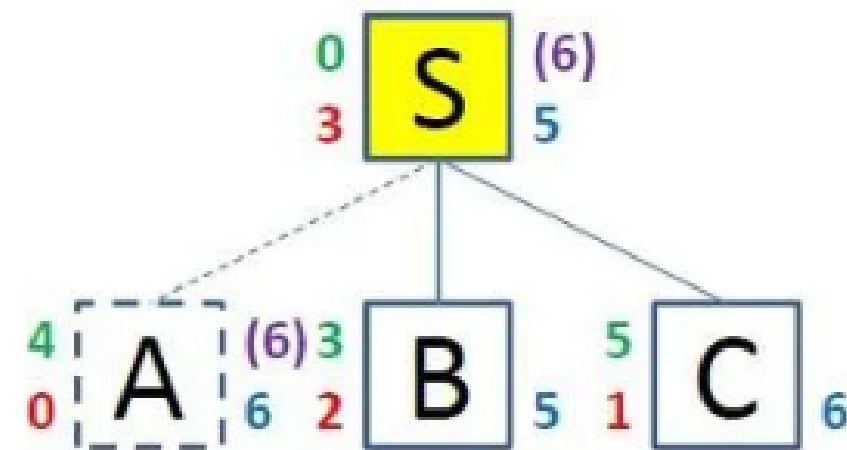
All children are
explored (update)

Adjust f-values

Image 10: Update The Root Value ("S" node) To The Remembered Value

SIMPLIFIED MEMORY BOUNDED A*

SMA* by Example



Generate children
(One by one)

Memory full

Image 11: Remove The Goal Leading Node

SIMPLIFIED MEMORY BOUNDED A*

SMA* by Example

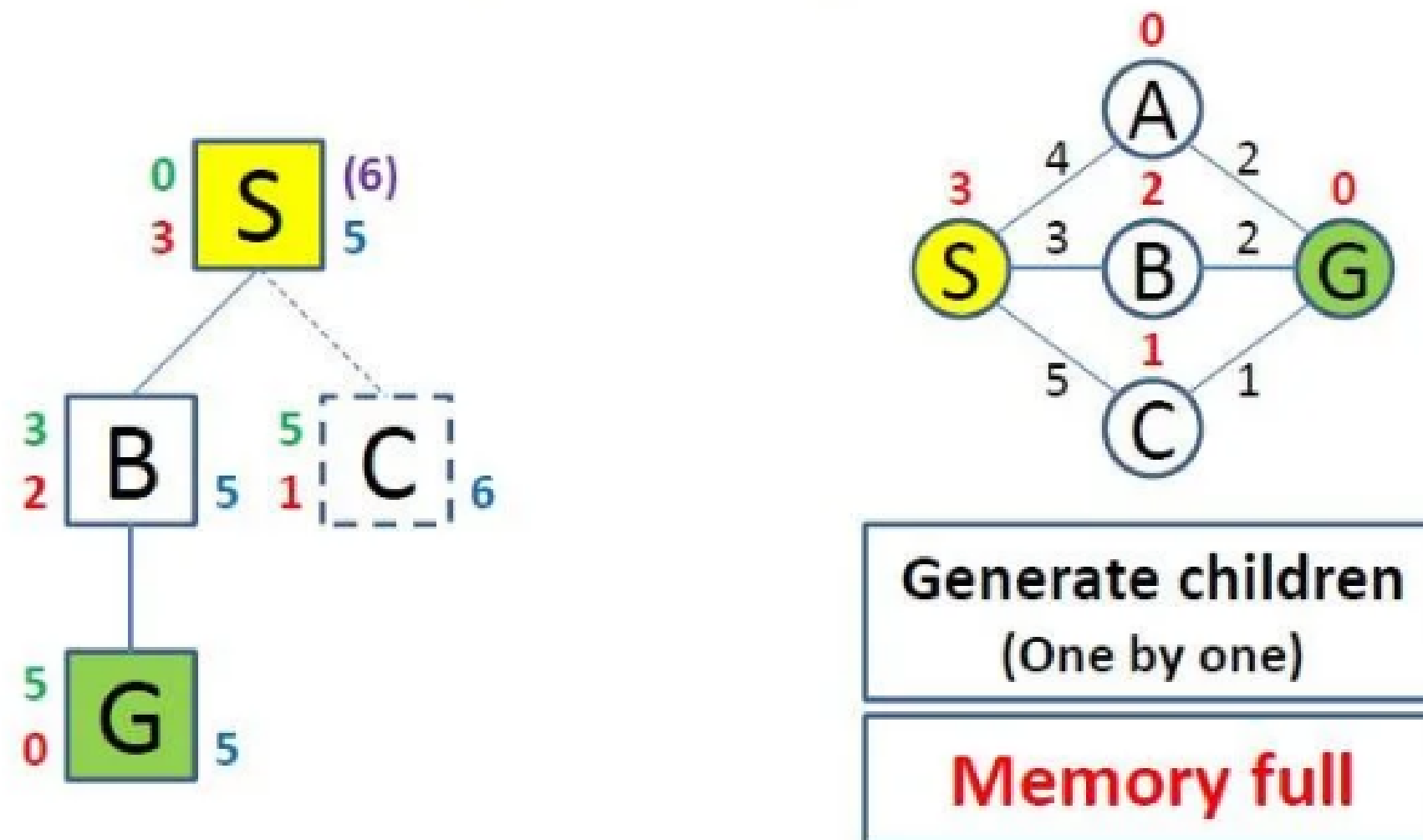


Image 12: Reaching The Goal For The Lowest Total Cost

bye