Most suitable search algorithm:

a) *“very large search space”*: DFS, IDS, or A* - not BFS
“large branching factor” : DFS or IDS
“possibly infinite paths” : not DFS
“no heuristic function” : not A* et al
“minimum no. of states” : optimal, so BFS or IDS

→ Iterative Deepening Search (IDS) is best

b) *“lots of cycles”* : not DFS
“varying costs” : UCS or A* - not BFS, DFS
“no heuristic function” : not A* et al
“shortest path” : optimal, so UCS or A*

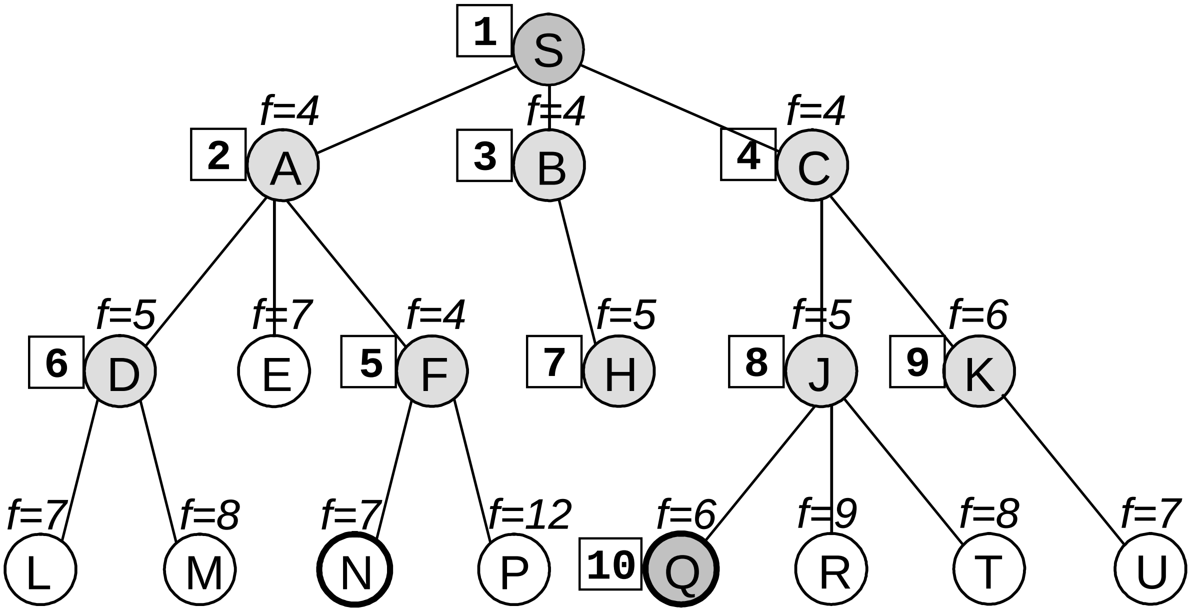
→ Uniform Cost Search (UCS / Dijsktra)

c) *“fixed depth tree”* : DFS, others
“goals at the bottom” : DFS - not BFS or IDS
“heuristic function” : Greedy Best First, not DFS
“find any goal quickly” : not opt., DFS or greedy
Best First

→ Greedy Best First Search is best

note: local search algorithms such as Hill-climbing
may fail to find a goal (no backtracking)

A search, solution and performance:*



queue

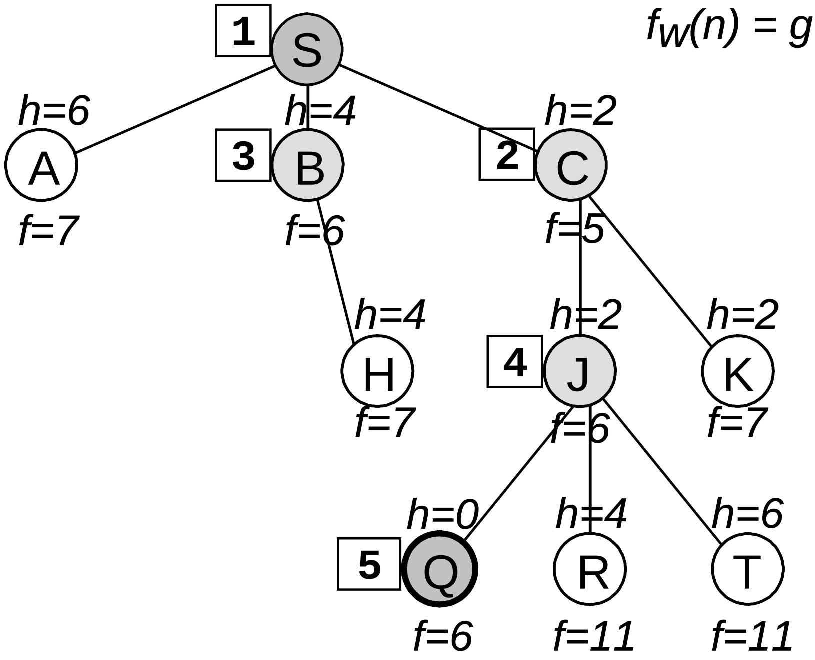
1. **S** (0+3=3)
2. **A** (1+3=4), **B** (2+2=4) , **C** (3+1=4)
3. **B**, **C**, **F** (3+1=4), **D** (4+1=5), **E** (4+3=7)
4. **C**, **F**, **D**, **H** (3+2=5), **E**
5. **F**, **D**, **H**, **J** (4+1=5), **K** (5+1=6), **E**
6. **D**, **H**, **J**, **K**, **E**, **N** (7+0=7), **P** (7+5=12)
7. **H**, **J**, **K**, **E**, **N**, **L** (5+2=7), **M** (7+1=8), **P**
8. **J**, **K**, **E**, **N**, **L**, **M**, **P**
9. **K**, **Q** (6+0=6), **E**, **N**, **L**, **M**, **T** (5+3=8), **R** (7+2=9), **P**
10. **Q**, **E**, **N**, **L**, **U** (6+1=7), **M**, **T**, **R**, **P**

nodes generated: 18
nodes expanded: 10
optimal solution

nearly exhaustive search (!)
ill-guided → poor heuristics
(optimistic, misleading)

Weighted A* search, solution and performance:

$f_W(n) = g(n) + 2 \times h(n)$



queue

1. **S** (0+6=6)
2. **C** (3+2=5), **B** (2+4=6), **A** (1+6=7)
3. **B**, **J** (4+2=6), **A**, **K** (5+2=7)
4. **J**, **A**, **K**, **H** (3+4=7)
5. **Q** (6+0=6), **A**, **K**, **H**, **R** (7+4=11), **T** (5+6=11)

nodes generated: 10 half(!) *well-guided search* → much improved heuristics

nodes expanded: 5

w-A* – pros: faster, complete

cons: not optimal (no guarantee)

increase *w*? faster yet, less and less optimal (still better than greedy search!)