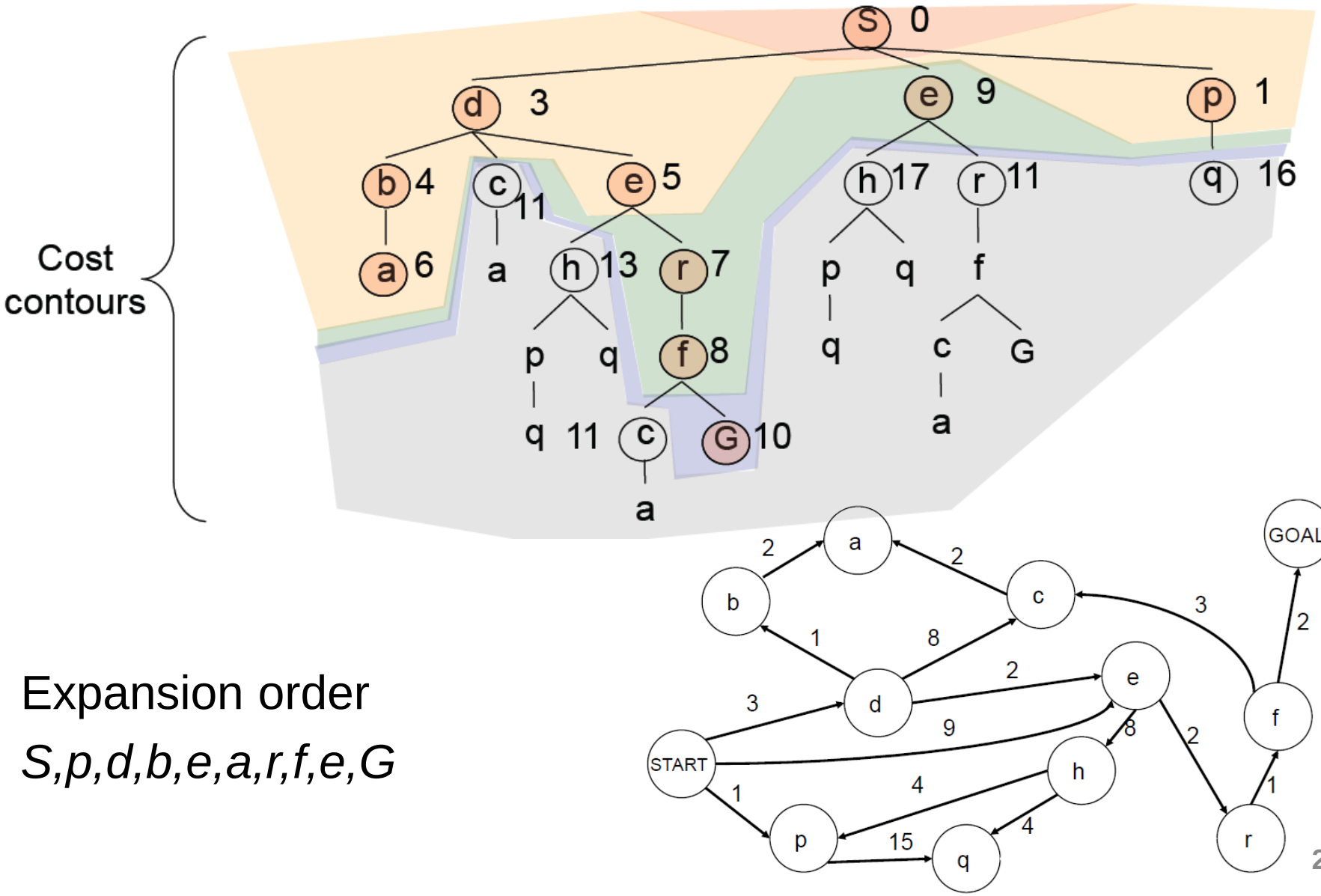
Uniform-cost search (UCS)

```
function UNIFORM-COST-SEARCH(problem) returns a solution, or failure
  node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  frontier ← a priority queue ordered by PATH-COST, with node as the element
  explored ← an empty set
  loop do
    if EMPTY?( frontier) then return failure
    node ← POP(frontier) /* chooses the lowest-cost node in frontier */
    if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
    add node.STATE to explored
    for each action in problem.ACTIONS(node.STATE) do
      child ← CHILD-NODE(problem, node, action)
      if child.STATE is not in explored or frontier then
        frontier ← INSERT(child, frontier)
      else if child.STATE is in frontier with higher PATH-COST then
        replace that frontier node with child
```

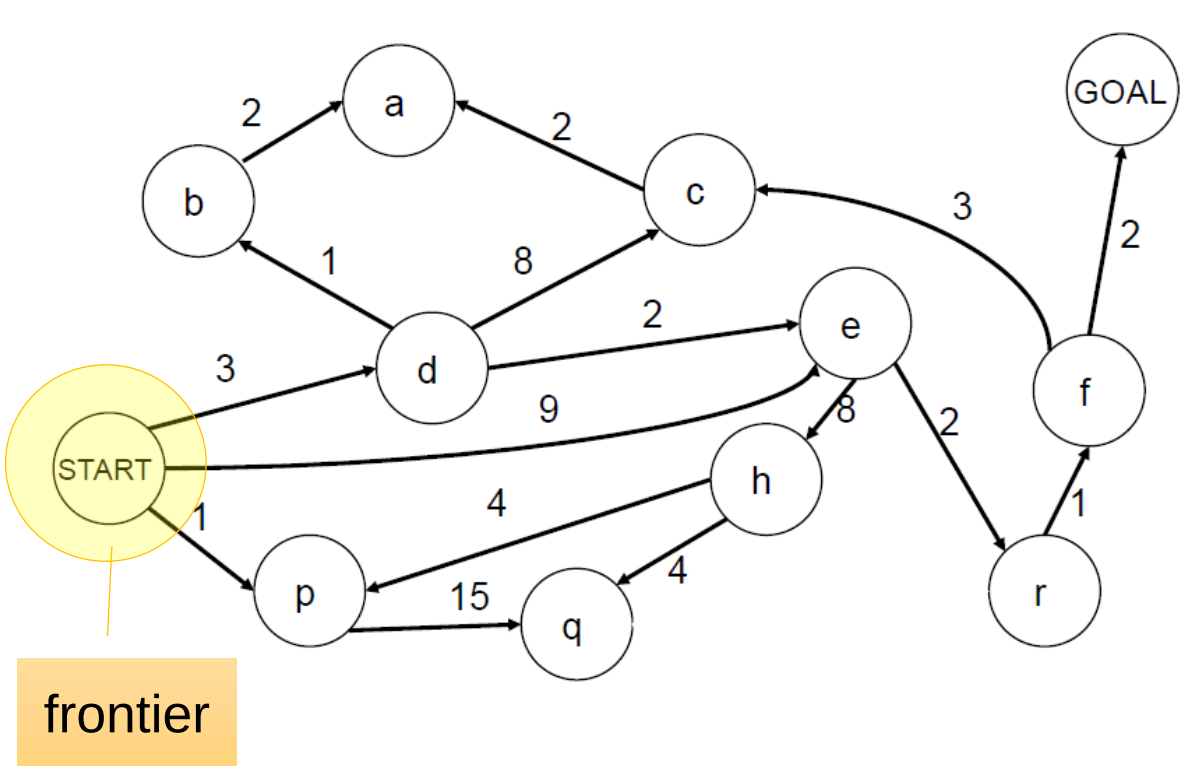
Uniform-cost search



Expansion order

S,p,d,b,e,a,r,f,e,G

Uniform-cost search: An example



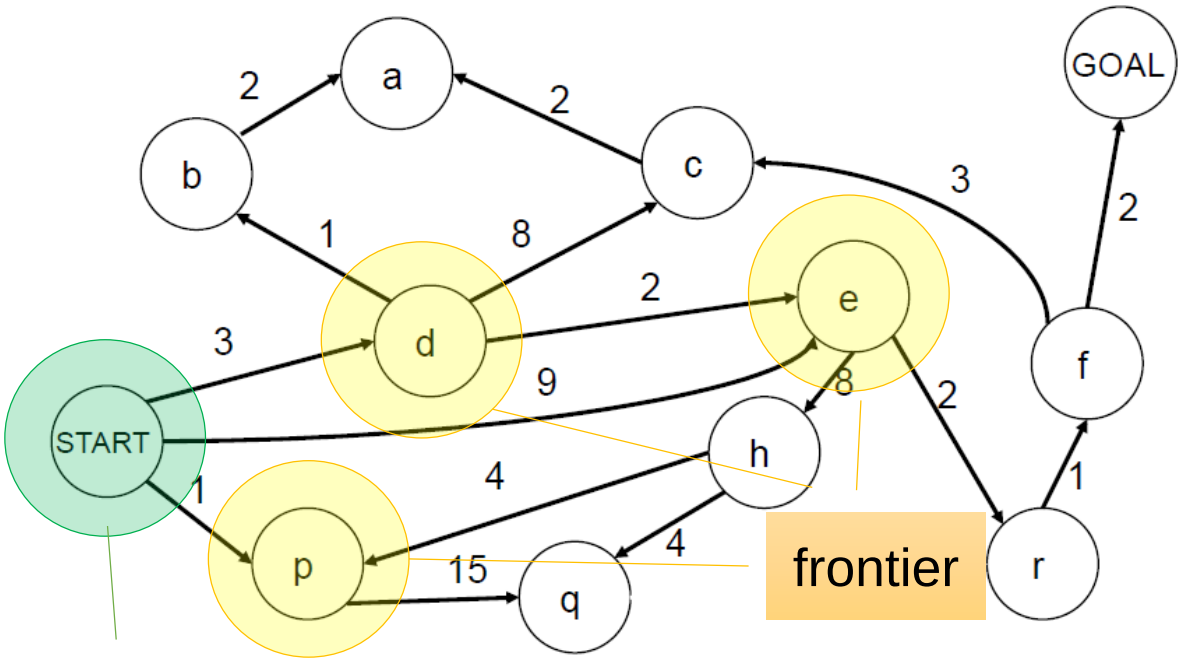
frontier

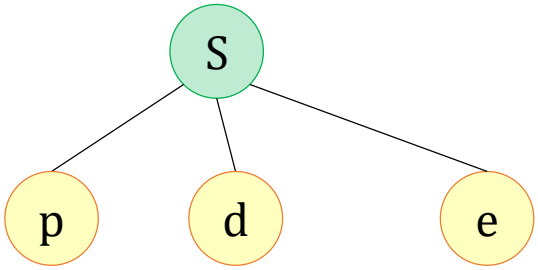
S

PQ = { (S:0) }

Search Tree

Uniform-cost search: An example



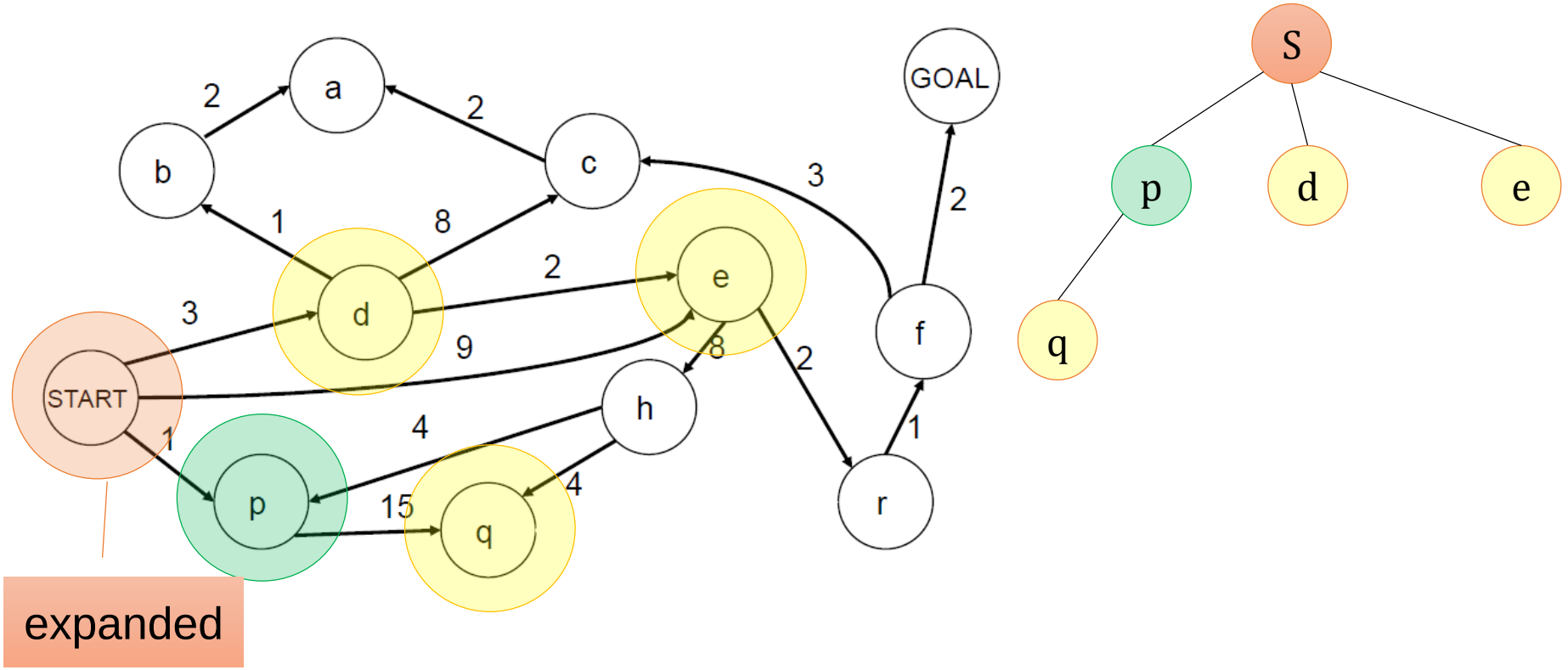


Selected for expansion

PQ = { (p:1), (d:3), (e:9) }

Search Tree

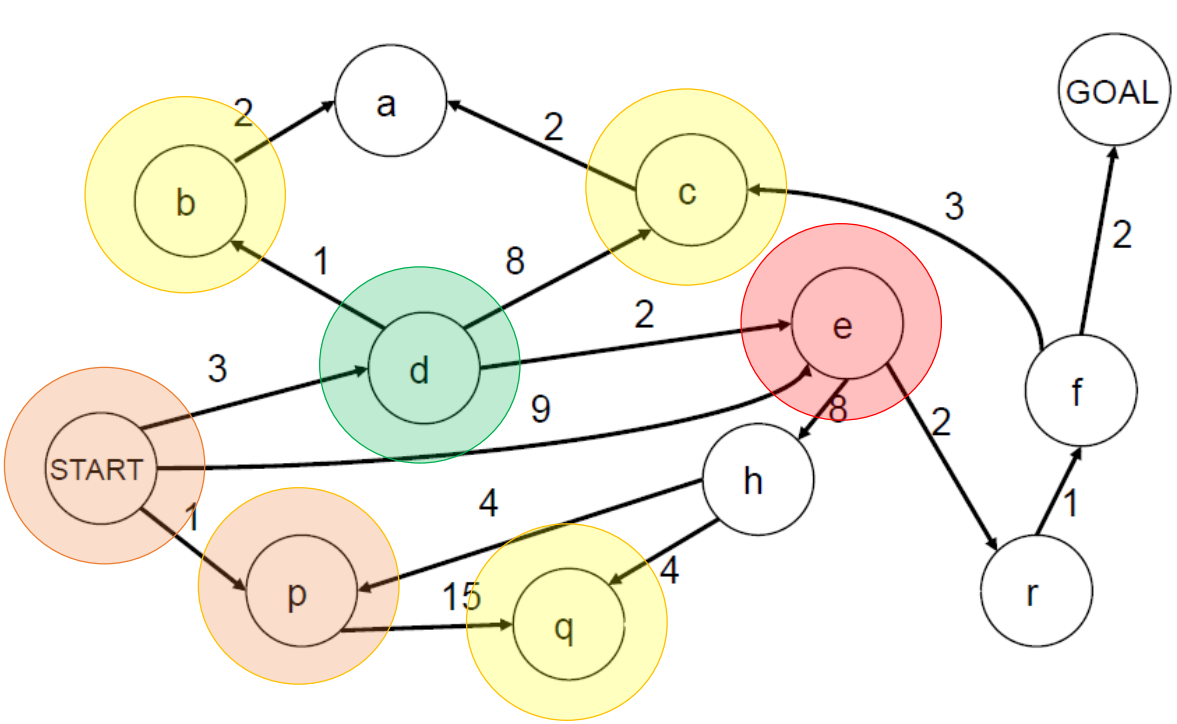
Uniform-cost search: An example

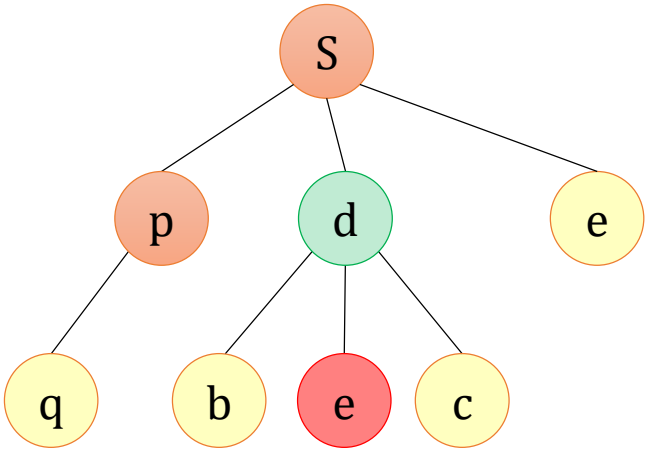


PQ = { (d:3), (e:9), (q:16) }

Search Tree

Uniform-cost search: An example



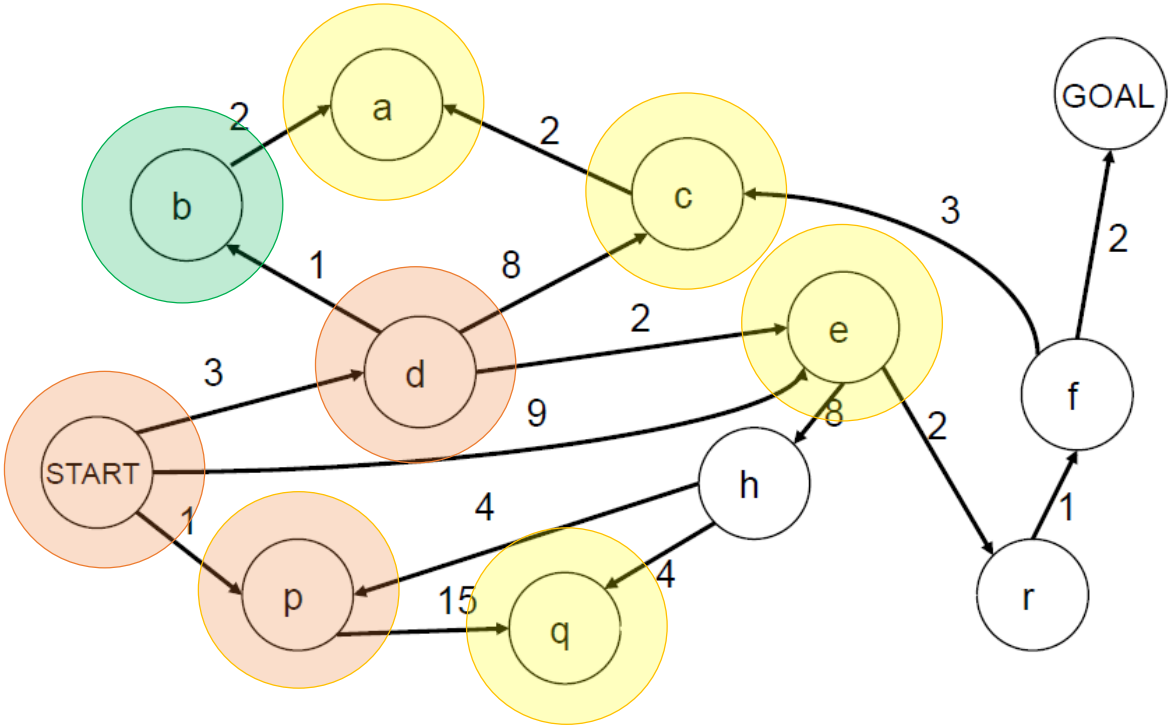


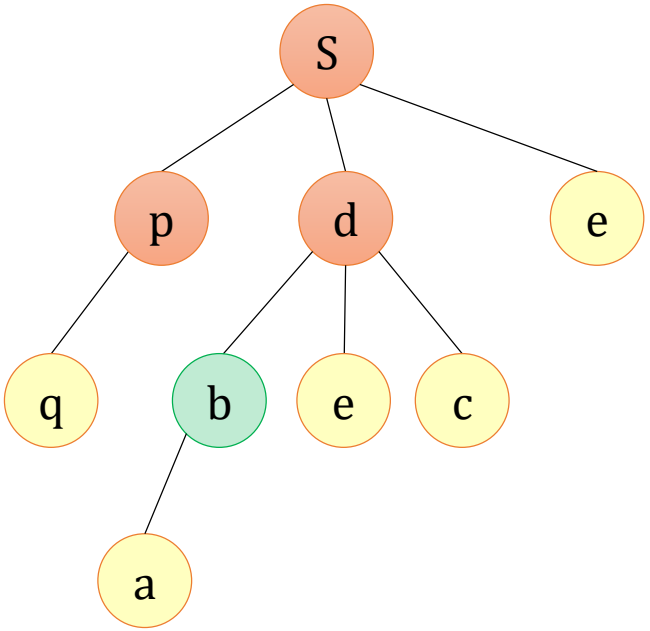
PQ = { (b:4), (e:5), (c:11), (q:16) }

Update path cost of *e*

Search Tree

Uniform-cost search: An example

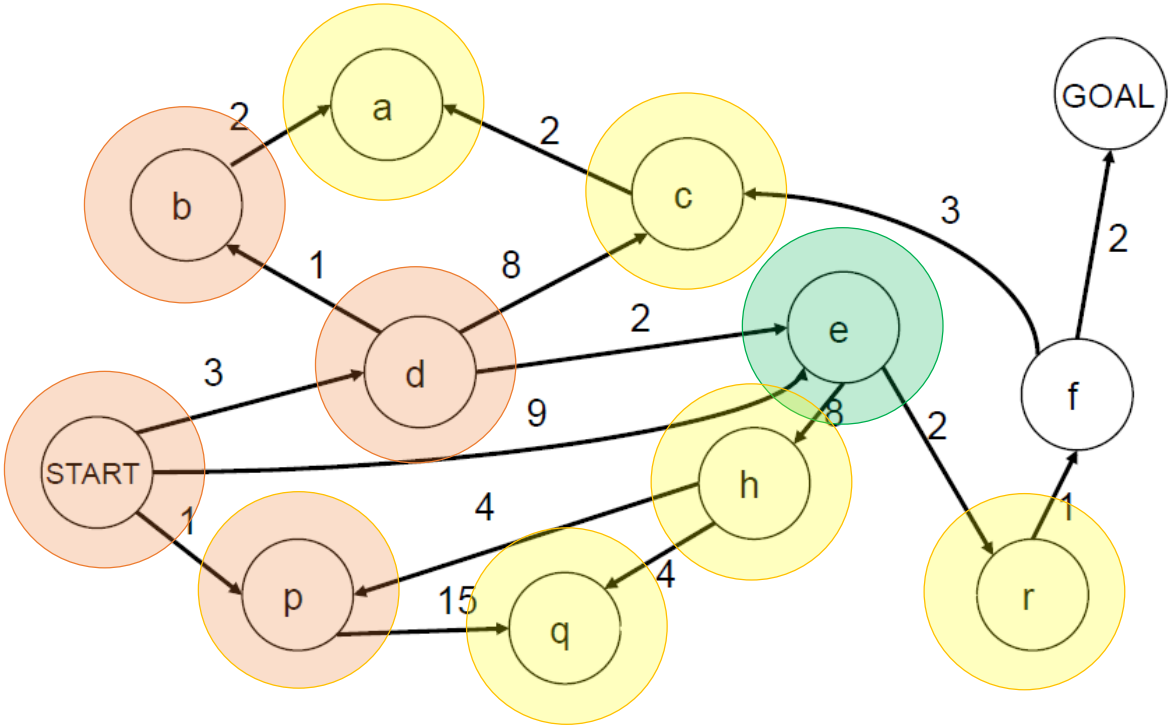


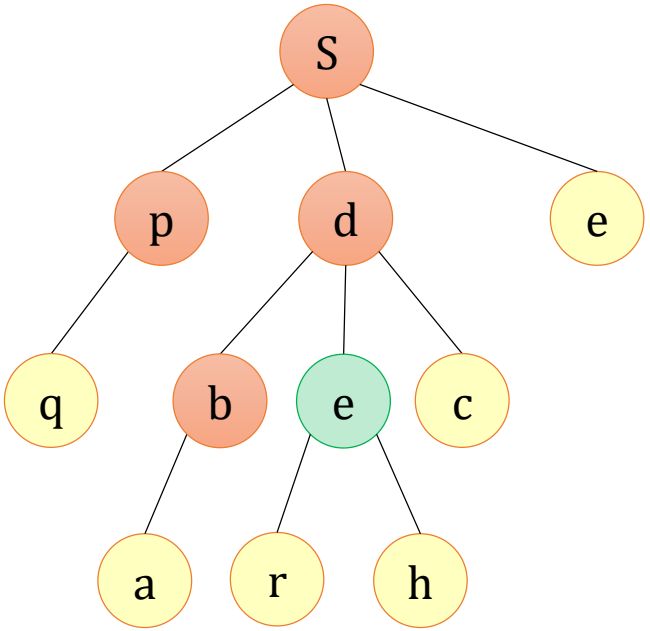


PQ = { (e:5), (a:6), (c:11), (q:16) }

Search Tree

Uniform-cost search: An example

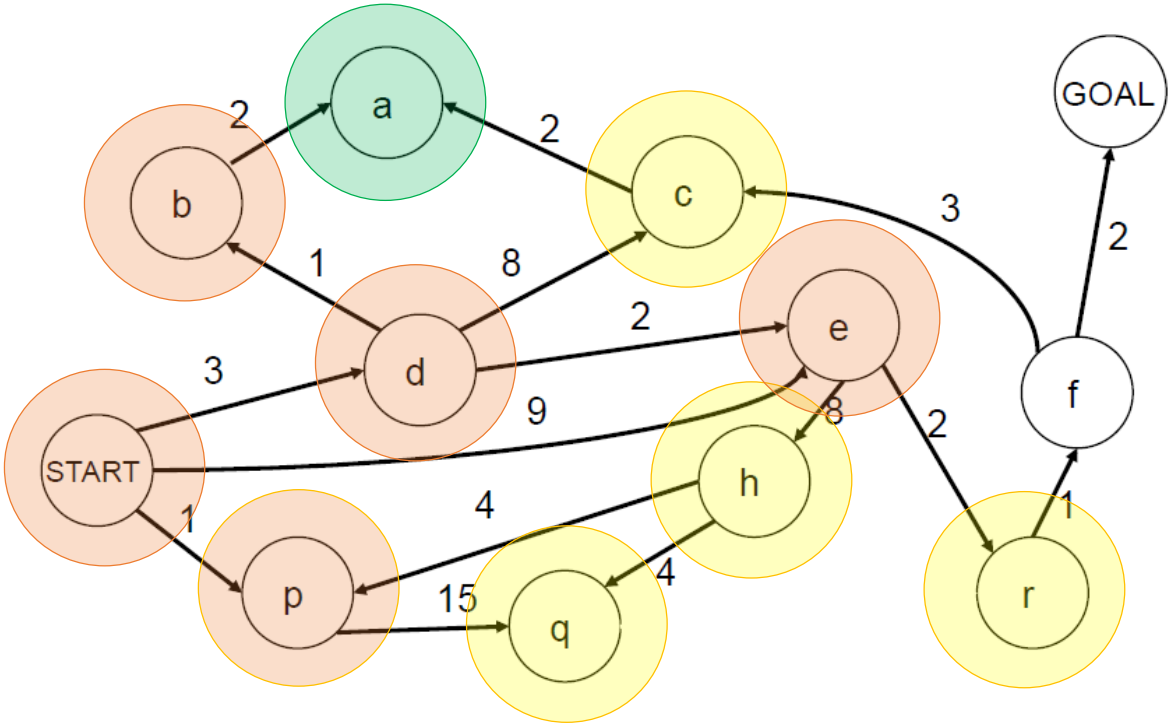


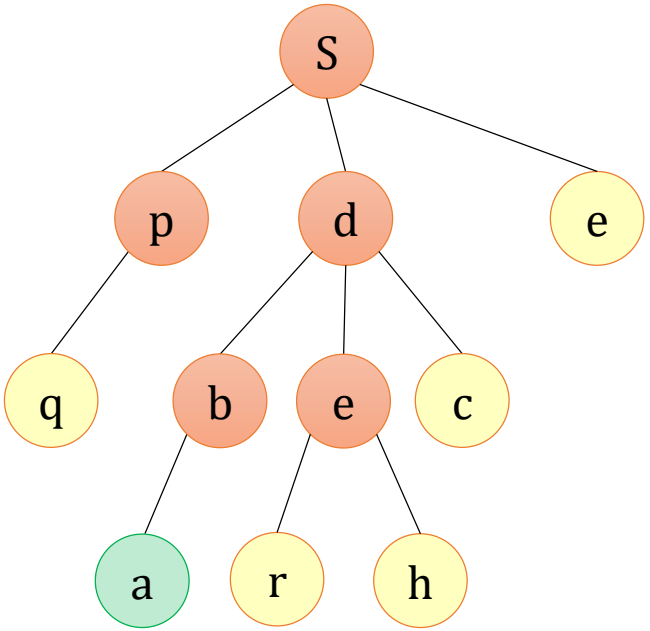


PQ = { (a:6), (r:7), (c:11), (h:13), (q:16) }

Search Tree

Uniform-cost search: An example

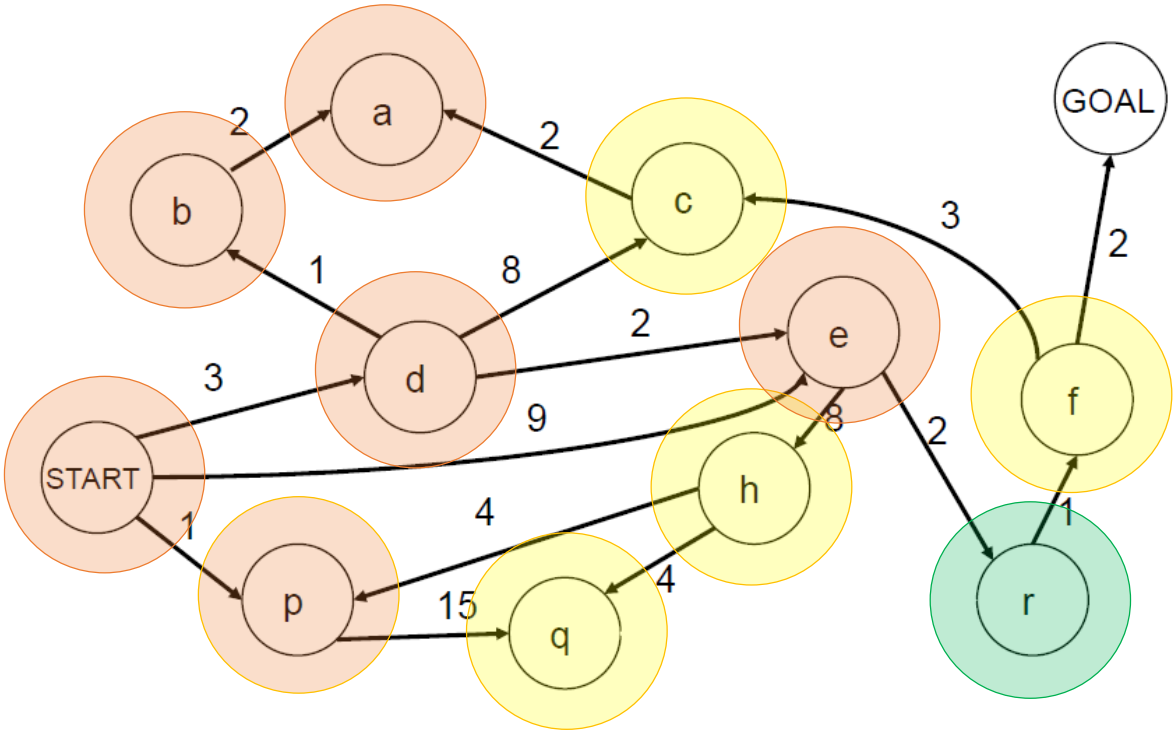


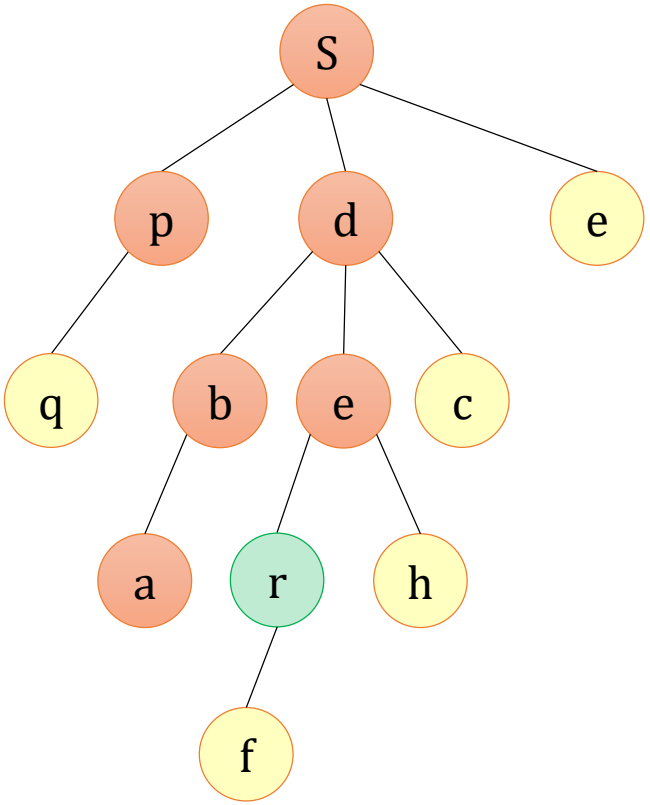


PQ = { (r:7), (c:11), (h:13), (q:16) }

Search Tree

Uniform-cost search: An example

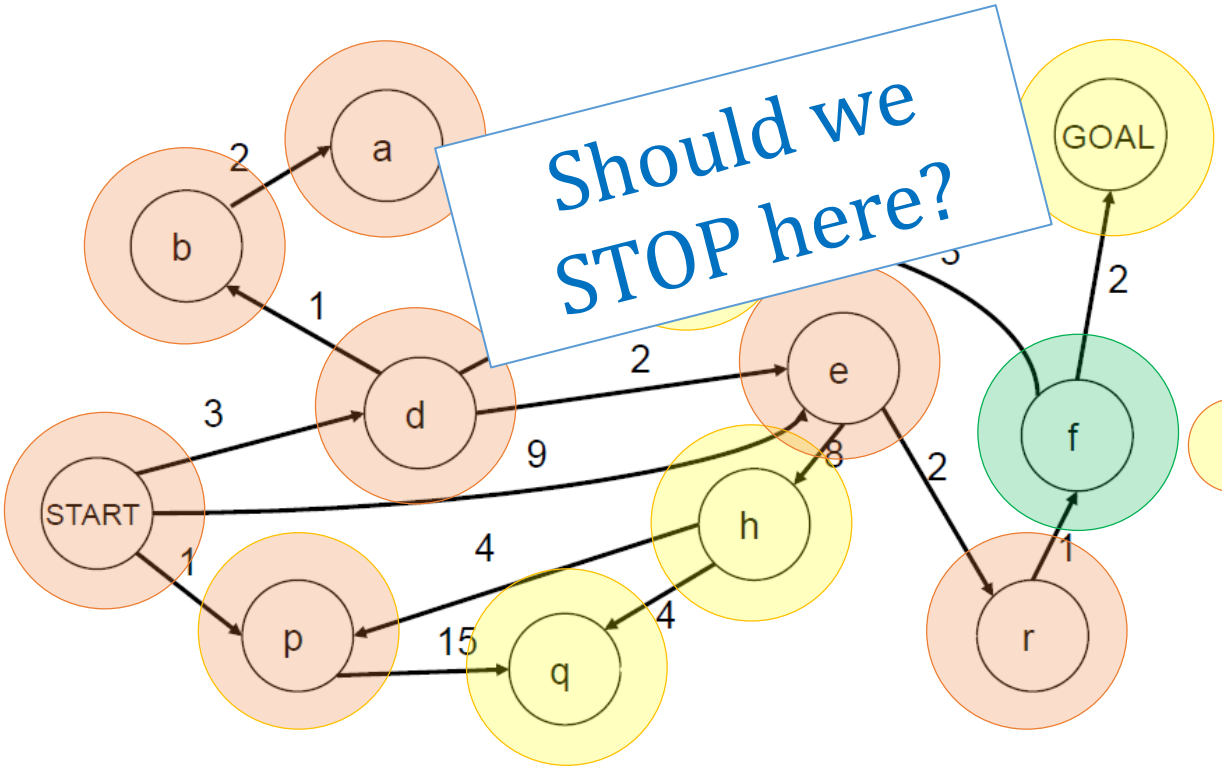


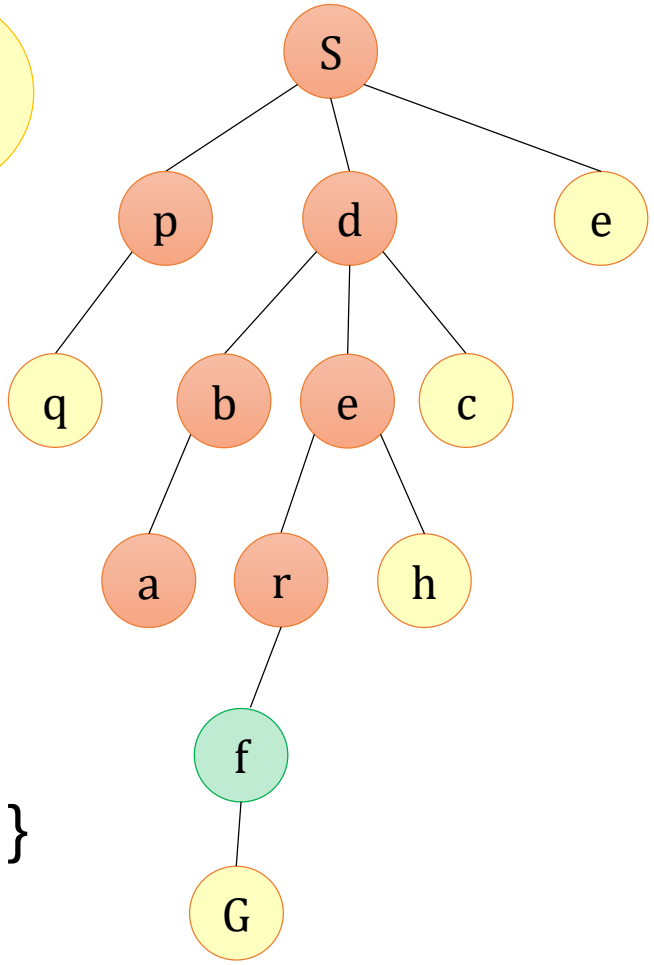


PQ = { (f:8), (c:11), (h:13), (q:16) }

Search Tree

Uniform-cost search: An example



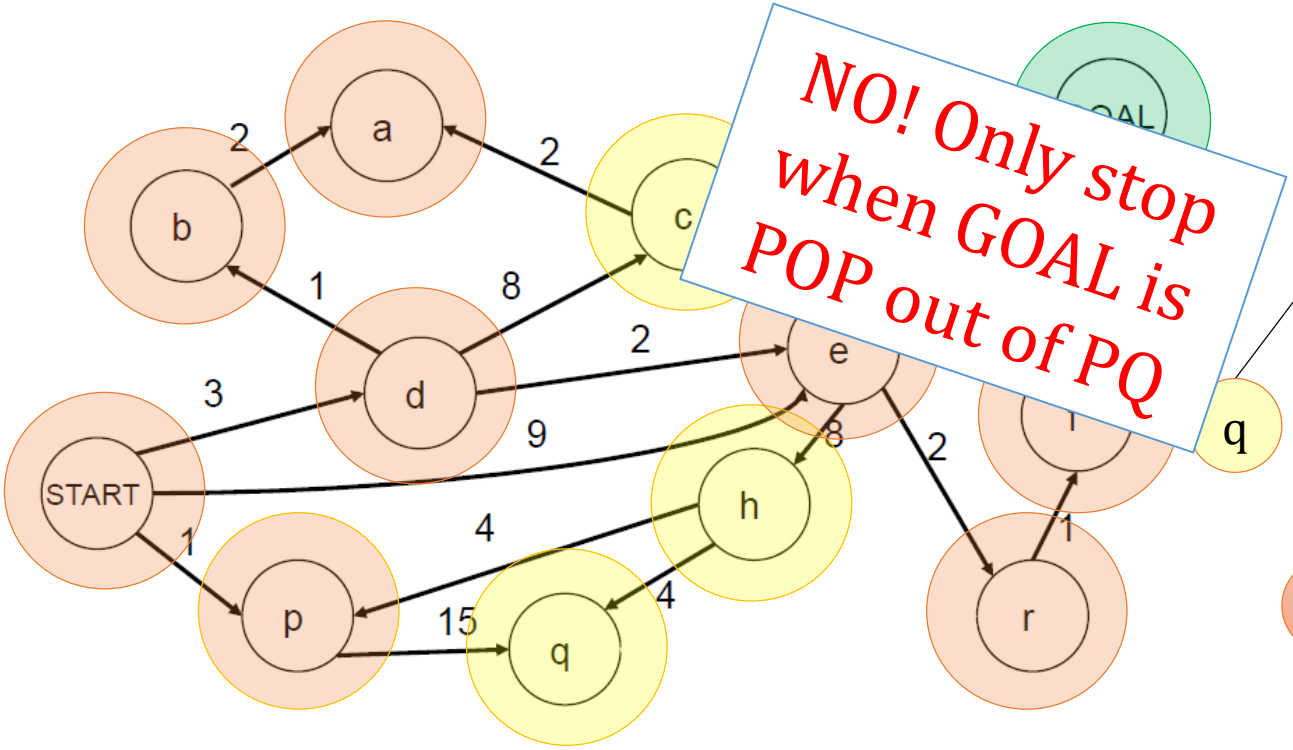


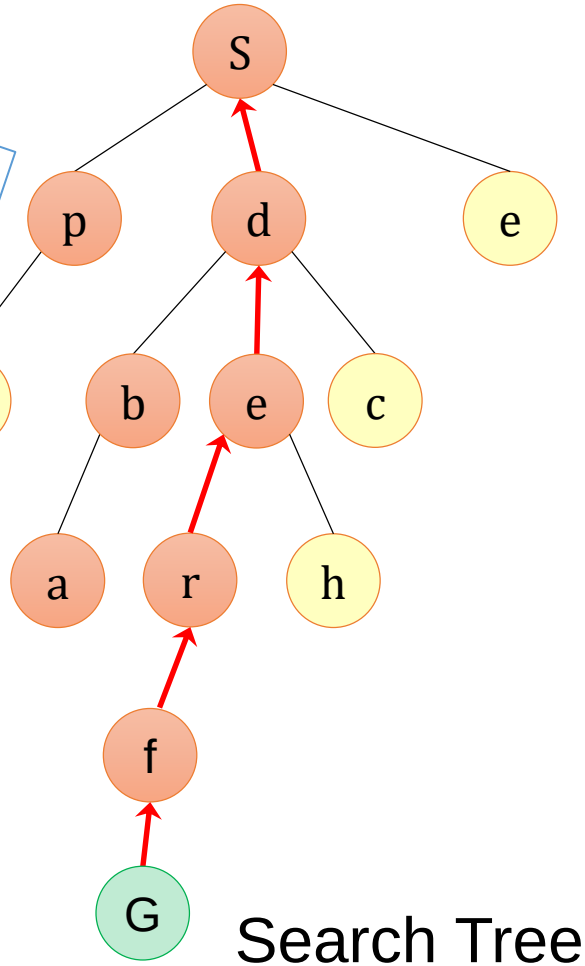
Search Tree

PQ = { (G:10), (c:11), (h:13), (q:16) }

Not update path cost of *c*

Uniform-cost search: An example





Search Tree

PQ = { (c:11), (h:13), (q:16) }

Goal is taken out of PQ → STOP

Search path: S → d → e → r → f → G, cost = 10