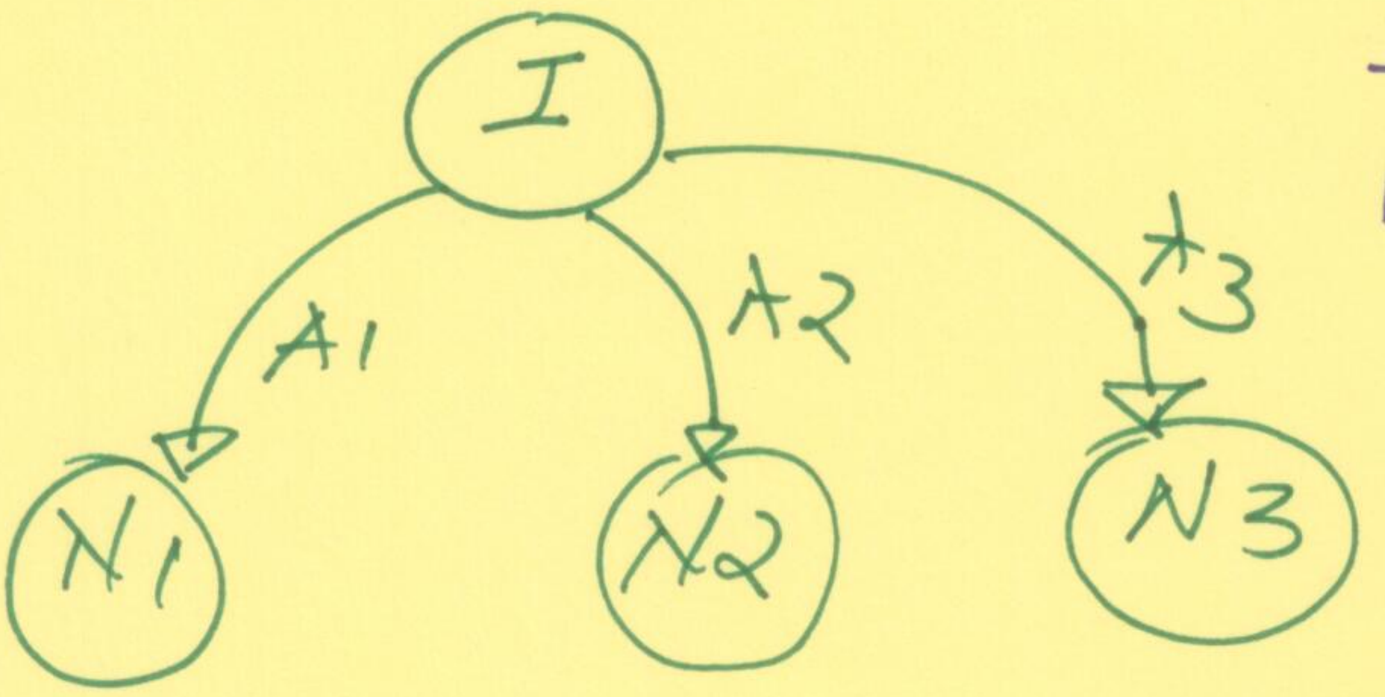
Search Trees



Step 1

Current Node

I

Frontier = {I}

Next node = ?

a) Remove I from frontier.

b) is I a GOAL? NO

c) Expand I

$A_1(I) = N_1$
$A_2(I) = N_2$
$A_3(I) = N_3$
} expanded nodes!

d) Add expanded nodes to Frontier

Frontier = {N1, N2, N3}

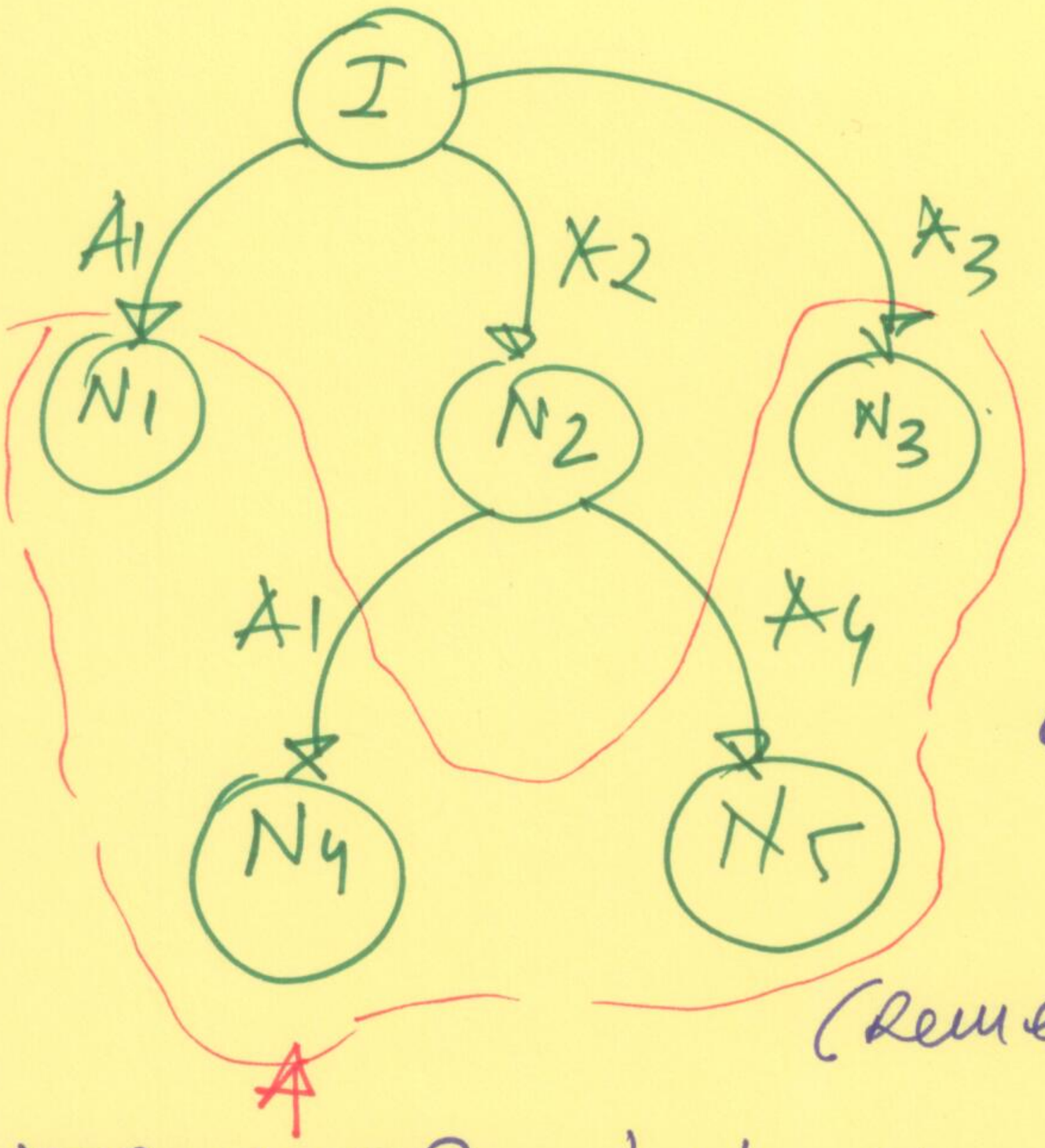
Step 2 a) Remove a node from Frontier

=> (N2) => New Frontier = {N1, N3}

Current node = N2

b) Is N2 a GOAL? NO

c) Expand N2



Step2 (cont)

$X_1(N_2) = N_4$

$A_4(N_2) = N_5$

d) Add expanded nodes to frontier (Remember New Frontier = {N1, N3} ?)

FRONTIER = { N1, N4, N3, N5 }

STEP 3 a) Remove a node from Frontier

=> N4 => Current Node

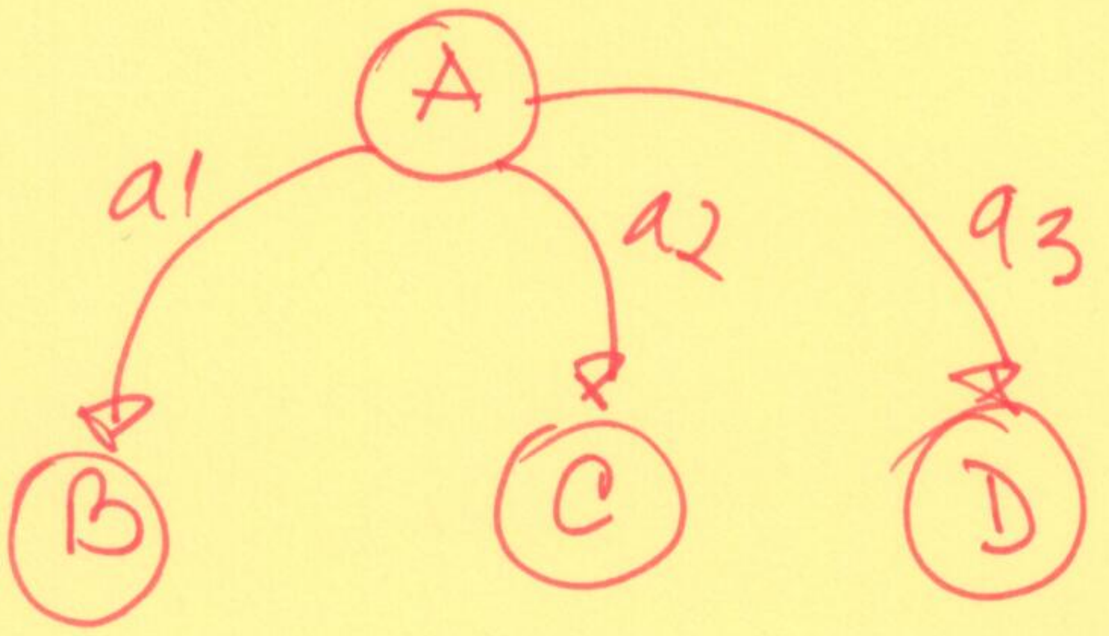
b) Is N4 = Goal. ?

YES

Return SOLUTION:

$I \xrightarrow{A_2} N_2 \xrightarrow{A_1} N_4 = GOAL$

Graph-search



Explored = {}
Frontier = {A}

Step 1 a) Remove A from Frontier

Current Node = A

b) IS A a GOAL? NO

c) ~~Expand~~ Explored = {A}

d) Expand A

a1(A) = B; a2(A) = C, a3(A) = D

- add expanded nodes to Frontier
ONLY if not in Frontier or Explored set

Frontier = {B, C, D}

Step 2 a) Remove a node from Frontier => B

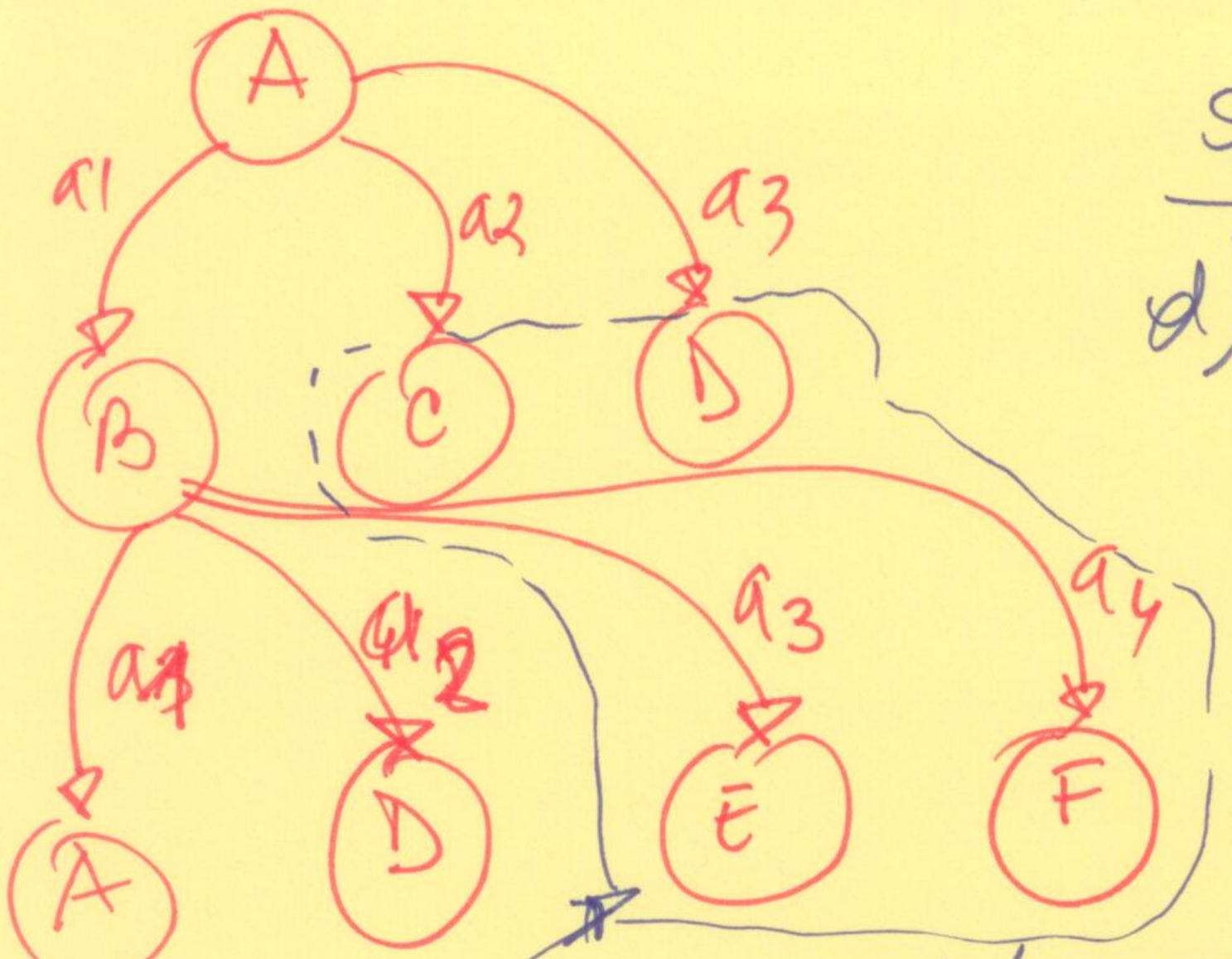
Current Node = B

New Frontier = {C, D}

b) IS B a GOAL? NO

c) Explored = {A, B}

d) Expand B = ...



Step2 Cout

d) Expand B

$a_1(B) = A$ X

$a_2(B) = D$ X

$a_3(B) = E$

$a_4(B) = F$

→ add expanded nodes in FRONTIER only if not in Frontier or Explored

Explored = {A, B}

we had Before Frontier = {C, D}

⇒ Frontier = {C, D, E, F}

Step 3 a) Remove a node from Frontier ⇒ C

Current Node = C → Frontier = {D, E, F}

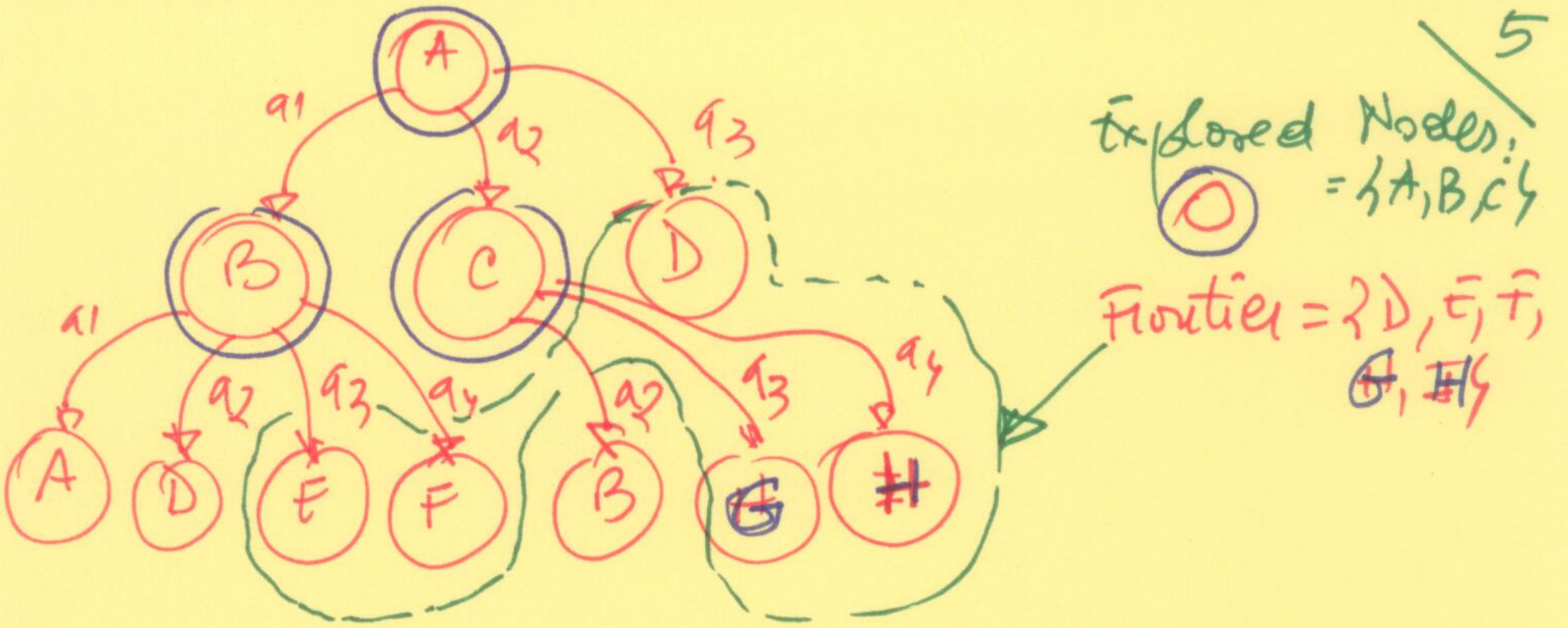
b) is C a GOAL ? No

c) Explored = {A, B, C}

d) Expand C : $a_2(C) = B$ X $a_3(C) = H$ ∪ $a_4(C) = H$ ∪

→ add expanded nodes in Frontier if not in Frontier or Explored

Frontier = {D, E, F, H}



Step 4 a) Remove a node from Frontier.

Current Node: G → Frontier = {D, E, F, H}

b) Is G a Goal? YES

Solution $A \xrightarrow{q_2} C \xrightarrow{q_3} G$

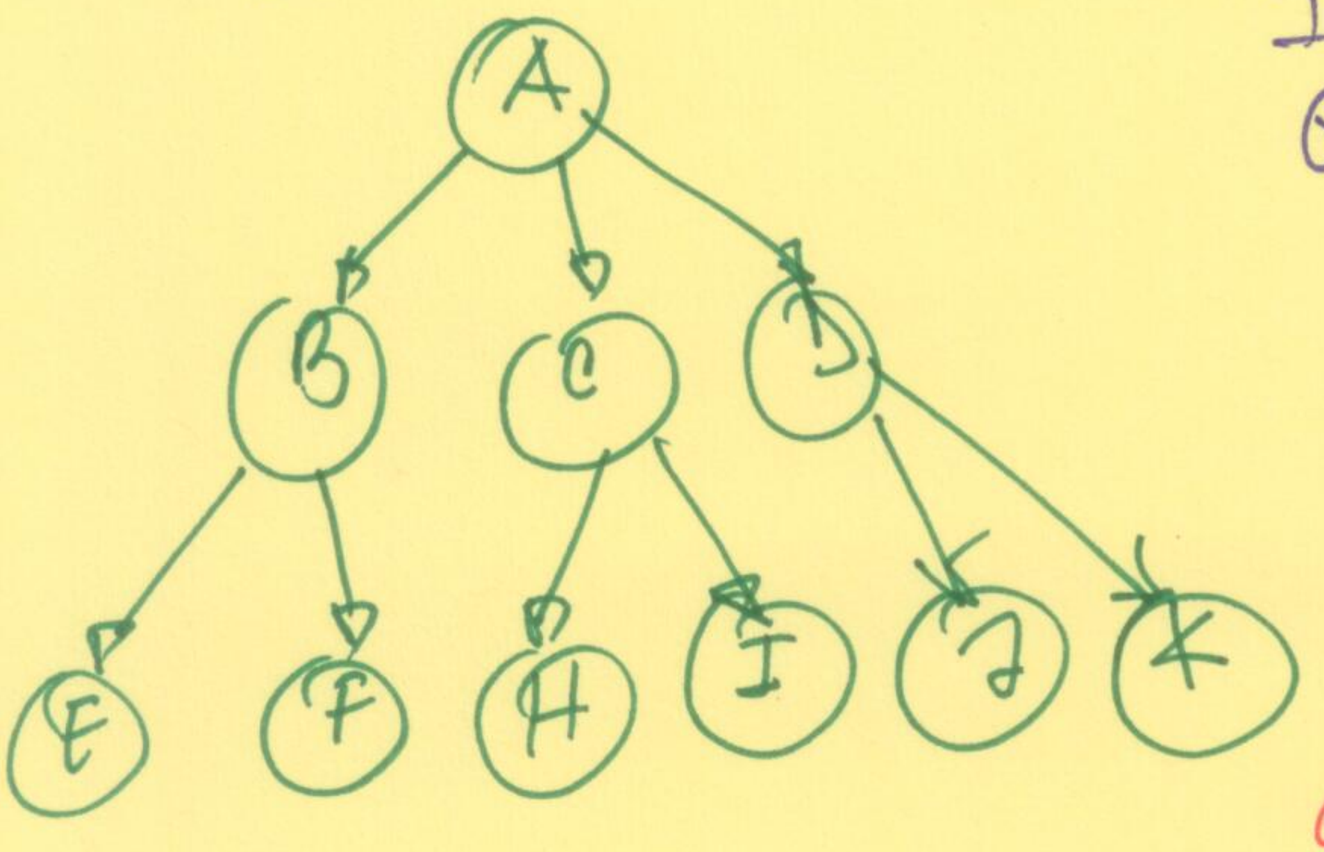
LESSON LEARNED:

Graph Search does not allow repeated states on Frontier or Expanded list

Uninformed Search Problems

1 Example of Breadth-First Search

a) Consider the following search tree:



Initial Node = A
Goal Node = J

Frontier = {A} FIFO queue
Explored = {}

Step 1 a) current node = A
b) children (A) = {B, C, D}
not in Frontier or Explored!
c) A is GOAL? NO
d) Frontier = {B, C, D}
Explored = {A}

Step 2 a) current node = B
Explored = {A, B}
Frontier = {C, D}
b) children (B) = {E, F} → not in Explored or Frontier!
c) is B a GOAL? NO
d) Frontier = {C, D, E, F}

Step 3 a) current node = C
Explored = {A, B, C}
Frontier = {D, E, F}
b) children (C) = {H, I} → not in ~~Expanded~~ Explored or Frontier!
c) is C a goal? NO
d) = Frontier = {D, E, F, H, I}

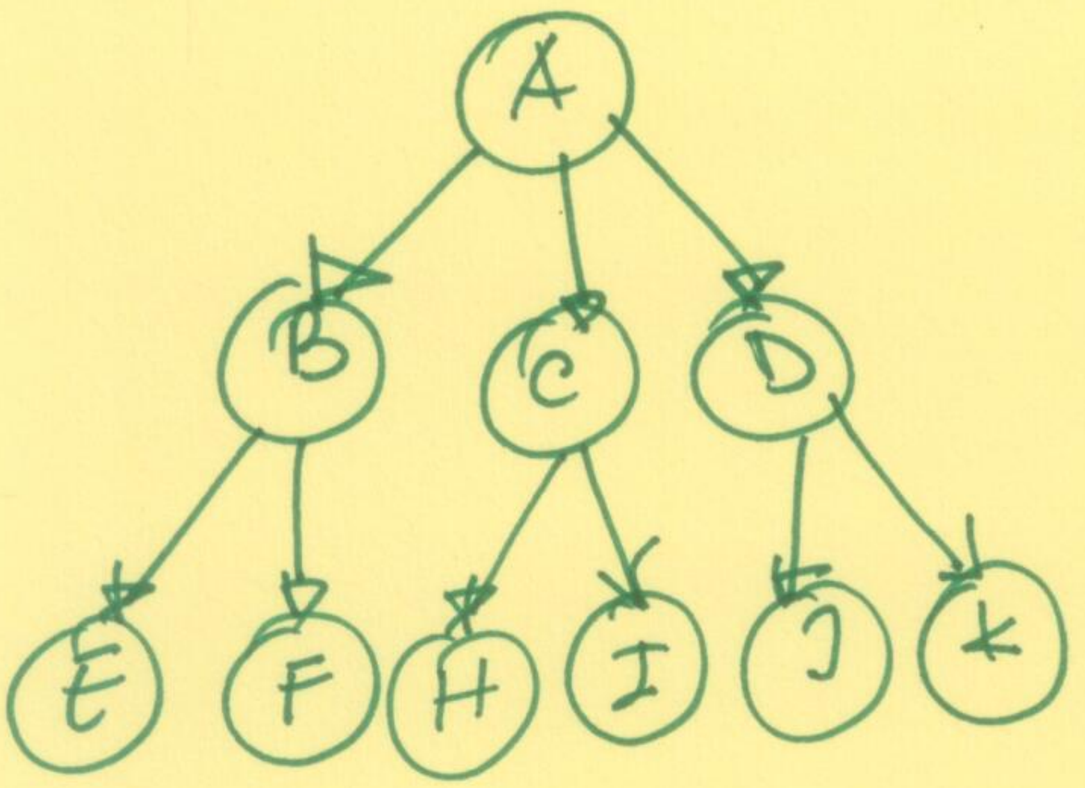
Step 4 a) Current node = D
Explored = {A, B, C, D} Frontier = {E, F, H, I
b) Children (D) = {I, K} → not in Explored or Frontier!
c) is D a Goal? NO
d) Frontier = {E, F, H, I, J, K}

Step 5 a) Current node = E
Explored = {A, B, C, D, E} Frontier = {F, H, I, J, K}
b) Children (E) = {}
c) is E a Goal? NO
d) Frontier = {F, H, I, J, K}

Step 6 a) Current node = F
Explored = {A, B, C, D, E, F}
Frontier = {H, I, J, K}
b) Children (F) = {}
c) is F a Goal? NO
d) Frontier = {H, I, J, K}

Step 7 a) Current node = H
Explored = {A, B, C, D, E, F, H} Frontier = {I, J, K}
b) Children (H) = {}
c) is H a Goal? NO
d) Frontier = {I, J, K}

Step 8 a) Current node = I
Explored = {A, B, C, D, E, F, H, I} Frontier = {J, K}
b) Children (I) = {}
c) is I a Goal? NO
d) Frontier = {J, K}



Step 9

a) Current node = J

Explored = {A, B, C, D, E, F, H, I, J}

Frontier = {K}

b) children(K) = {}

c) Is K a GOAL? YES

SOLUTION: A → D → J

How did I find it?

1. Which node expanded J?
(when did J first show up on the Frontier?)

↓

Step 6 => J is a child of D

2) Which node expanded D?
(when did D first show up on the Frontier?)

↓

Step 1 => D is a child of A

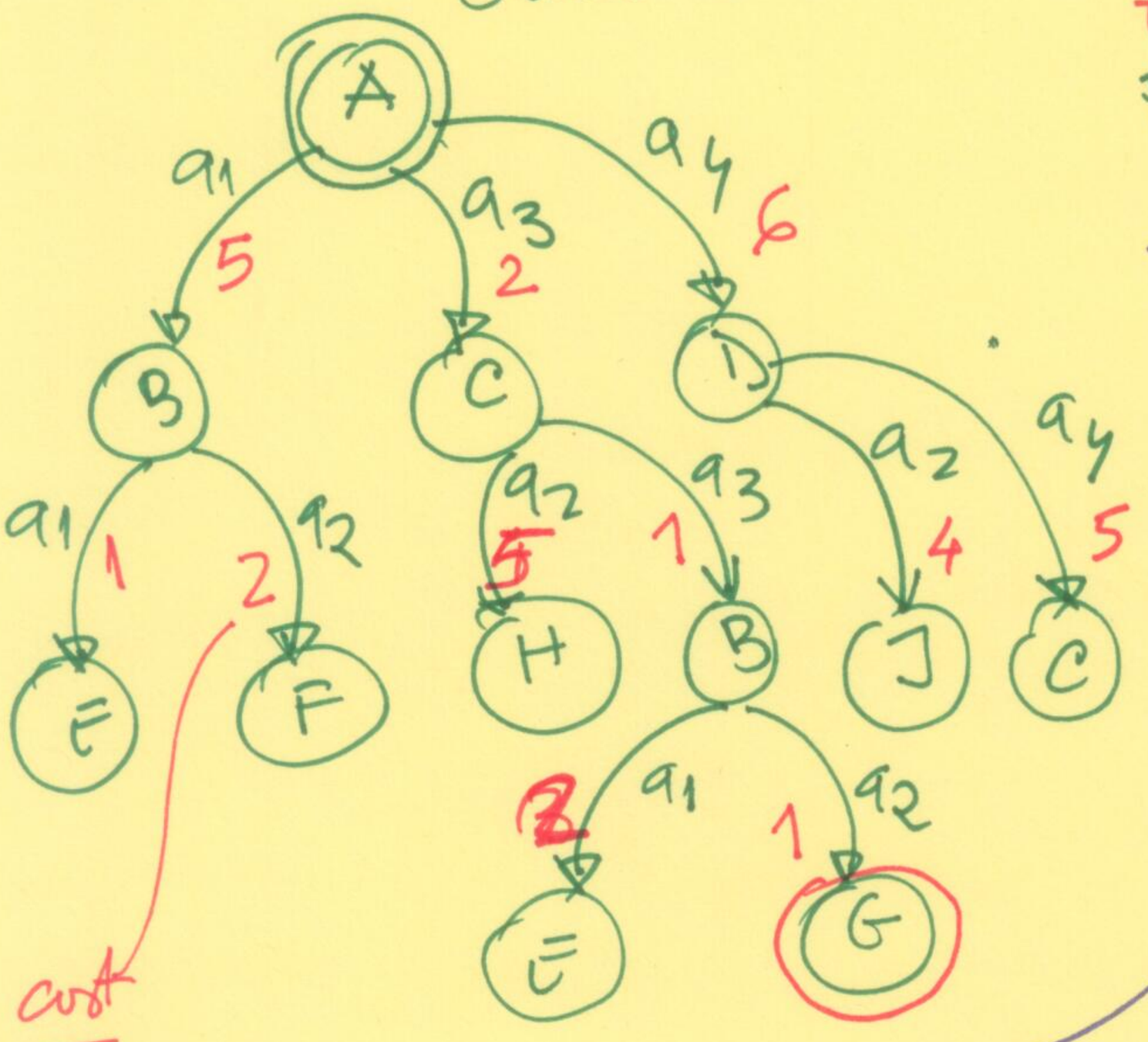
I can create the solution path:
A → D → J.

(A is the initial node)

2 Example of Uniform Search

We always have $f(n)$ = the path cost from initial node to node n

Problem: Actions = {a_1, a_2, a_3, a_4}
Initial state = A
Goal State = G.



Explored = {}
Frontier = {A}

Step 1 a) Current Node = A

b) is A the Goal? No

c) Explored = {A}
Frontier = {}

d) Consider for each available action the child-node

B = child(A, a_2)
$f(B) = 5$

C = child(A, a_3)
$f(C) = 2$

D = child(A, a_4)
$f(D) = 6$

* if B is not in Explored or Frontier
⇒ Frontier = {B 5}

* if C is not in Explored or Frontier
Frontier = {C 2, B 5}

if D is not in Explored or Frontier
Frontier = {C 2, B 5, D 6}

Step 2 a) Current Node = C $g(C)=2$

b) Is C a Goal? No

c) Explored = {A, C}

Frontier = { B D } / { 5 6 }

d) For each action → child (C, a2) = H ← not in Explored or Frontier

$g(H) = g(C) + 5 = 2+5 = 7$

Frontier = { B D H } / { 5 6 7 }

child (C, a3) = B → B is already in Frontier But 3 < 5 ⇒ replace it in Frontier

$g(B) = g(C) + 1 = 2+1 = 3$

⇒ Frontier = { B D H } / { 3 6 7 }

Step 3 a) Current Node = B $g(B)=3$

b) Is B a Goal? No

c) Explored = {A, B, C}

Frontier = { D H } / { 6 7 }

child (B, a1) = E $g(E) = g(B) + 2 = 3+2 = 5$ ← not in Explored or Frontier

Frontier = { E D H } / { 5 6 7 }

child (B, a2) = G $g(G) = g(B) + 1 = 3+1 = 4$ ← not in Explored or Frontier

Frontier = { G E D H } / { 4 5 6 7 }

Step 4 a) Current Node = G $g(G)=4$.

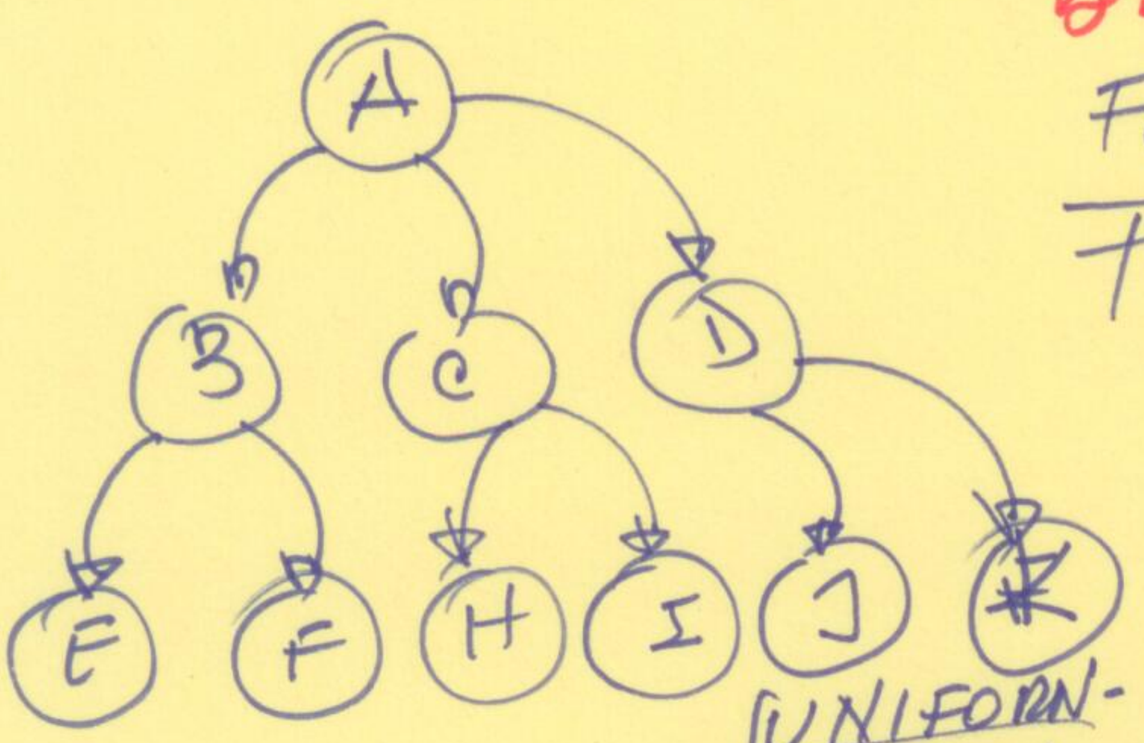
b) G is the Goal!

SOLUTION: A → C → B → G PATH COST = 4

Qualifying PhD exam Question

Prove that breadth-first search is a special case of uniform-cost search.

Answer If all step costs $c(n_1, a_i, n_2)$ are equal then $g(n) \propto \text{depth}(n)$ and uniform-cost search becomes breadth-first search.



UNIFORM-SEARCH

For example consider the illustrated search tree, with * ← initial node

Frontier = {A}
Explored = { }

Step 1 Current Node = A
is A the GOAL? NO
Explored = {A}
B = child (A) $\rho(B) = 1$
~~c = cb~~ Frontier = { B 1 }
C = child (A) $\rho(C) = 1$
Frontier = { B C 1 1 }
D = child (A) $\rho(D) = 1$
Frontier = { B C D 1 1 1 }

Step 2 Current Node = B
Explored = {A, B}
is B GOAL? NO
child (B) = E $\rho(E) = 1+2 = 3$
Frontier = { C D E 1 1 3 }
child (B) = F $\rho(F) = 3$
Frontier = { C D E F 1 1 3 3 }

Step 3 Current Node = C $p(C)=1$

Explored = {A, B, C}

is C a GOAL? NO.

child (C) = H $p(H) = 1+2 = 3$

Frontier = { D E F H / 1 3 3 3 }

child (C) = I $p(I) = 1+2 = 3$

Frontier = { D E F H I / 1 3 3 3 3 }

Step 4 Current Node = D $p(D) = 1$

Explored = {A, B, C, D}

Frontier = { E F H I J K / 3 3 3 3 3 3 }

Step 4 Current Node = E

Explored = {A, B, C, D, E} Frontier = { F H I J K / 3 3 3 3 3 }

Step 5 Current Node = ~~H~~

Explored = {A, B, C, D, E, F} Frontier = { H I J K / 3 3 3 3 }

Step 6 Current Node = H

Explored = {A, B, C, D, E, F, H} Frontier = { I J K / 3 3 3 }

Step 7 Current Node = I

Explored = {A, B, C, D, E, F, H, I} Frontier = { J K / 3 3 }

Step 8 Current Node = J

Explored = {A, B, C, D, E, F, H, I, J} Frontier = { K / 3 }

Step 9 Current Node = K GOAL!

SOLUTION : A → D → K PATH COST = 3

The content of the Frontier and Explored-list are the same at each step with those obtained by Breadth-First SEARCH