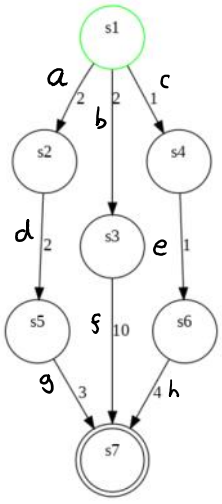
Week 2 T1

Wednesday, 3 August 2022 9:33 AM

State Model $\mathcal{S}(P)$:

- finite and discrete state space S
- a known initial state $s_0 \in S$
- a set $S_G \subseteq S$ of goal states
- actions $A(s) \subseteq A$ applicable in each $s \in S$
- a deterministic transition function $s' = f(a, s)$ for $a \in A(s)$
- positive action costs $c(a, s)$

→ A **solution** is a sequence of applicable actions that maps s_0 into S_G, and it is **optimal** if it minimizes **sum of action costs** (e.g., # of steps)



$S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$

$s_0 = s_1$

$S_G = \{s_7\}$

$A = \{A(s_1) = \{a, b, c\}, A(s_2) = \{d\}, A(s_3) = \{f\}, A(s_4) = \{e\}, A(s_5) = \{g\}, A(s_6) = \{h\}, A(s_7) = \emptyset\}$

$T = \{f(s_1, a) = s_2, f(s_1, b) = s_3, f(s_1, c) = s_4, f(s_2, d) = s_5, f(s_3, f) = s_7, f(s_4, e) = s_6, f(s_5, g) = s_7, f(s_6, h) = s_7\}$

$C = \{c(a) = 2, c(b) = 2, c(c) = 1, c(d) = 2, c(e) = 1, c(f) = 10, c(g) = 3, c(h) = 4\}$

General Search Strategy

Open Set : stores nodes yet to be explored

Closed Set : stores explored nodes

- Start with empty tree, empty open and closed sets
- Create initial node and add it to open set
- Repeat :
  1) Choose node from open set to expand
  2) Remove node from open set (add to tree)
  3) Check if goal state (terminate if goal)
  4) Generate child nodes and add them to open set (typically ignore nodes if they are in closed set)
  5) Add node to closed set

Different Search Strategies determine choice order

BrFS

- Chooses earliest node
- FIFO first in first out
- Use queue for open set

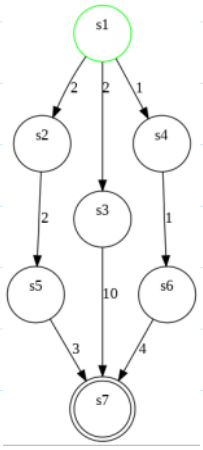
DFS

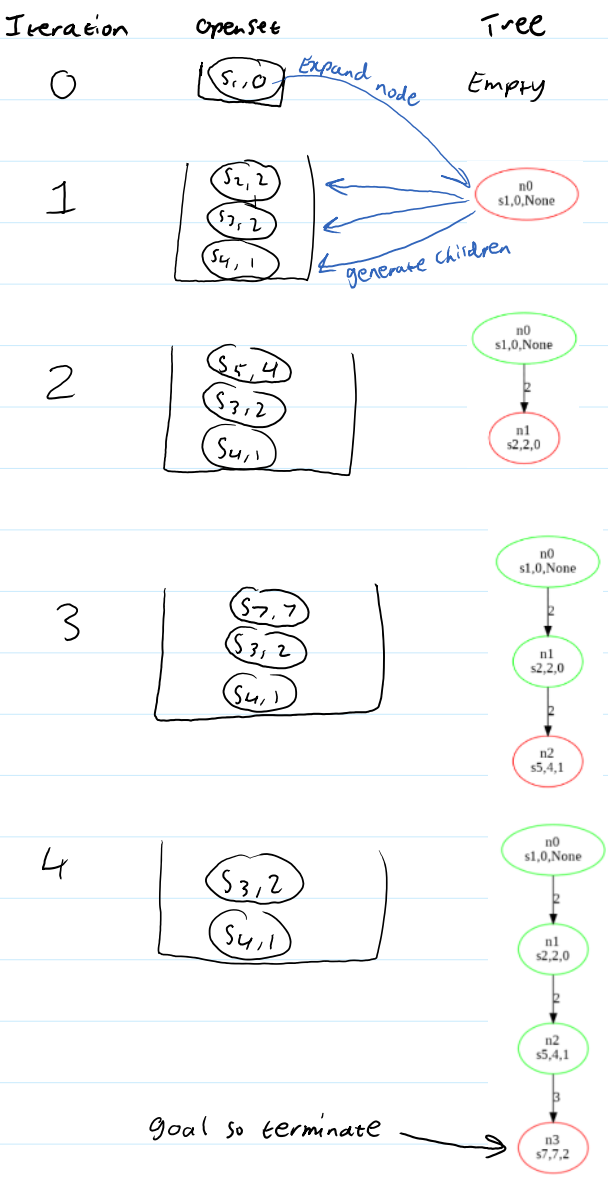
- Chooses latest node
- LIFO last in first out
- Use stack for open set

ID

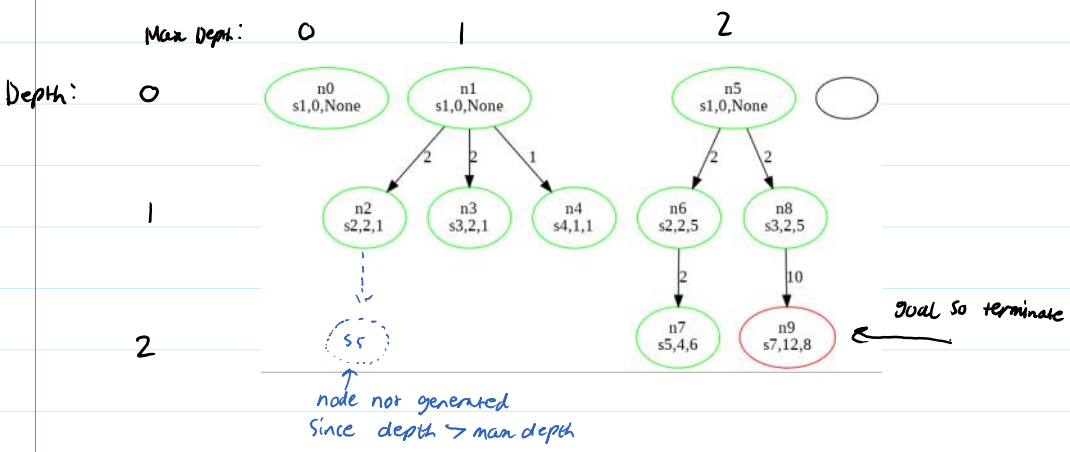
- Repeatedly performs depth limited depth first search
- DLDFS same as DFS, but only generates children below a max depth.
- After each iteration, max depth increases by 1
- Combines BrFS and DFS

DFS Example





ID Example


Max Depth: 0 1 2
Depth: 0
1
2
n0
s1,0,None
n1
s1,0,None
n5
s1,0,None
2
2
1
2
2
n2
s2,2,1
n3
s3,2,1
n4
s4,1,1
n6
s2,2,5
n8
s3,2,5
2
10
s5
n7
s5,4,6
n9
s7,12,8
Goal so terminate
node not generated
Since depth > max depth