

COMP90054 Workshop 2

Geye Guo

Recap

- Blind search: only use basic search algorithm (BFS, DFS, ID)

	Complete	Optimal	Time Complexity	Space Complexity
BFS	T	T*	$O(b^d)$	$O(b^d)$
DFS	F	F	infinity	$O(b^*d)$
ID	T	T*	$O(b^d)$	$O(b^*d)$

b = branching factor, d = depth of the optimal path

- Heuristic Search: additionally use the heuristic function to estimate the remaining cost (distance) to the goal state

A few notations for heuristic search

- $s, s', a, c(a)$
- $n = \langle s, f(n), g(n), n_{parent} \rangle$
- $h \leftrightarrow h(s), h^* \leftrightarrow h^*(s)$

- **Uniform cost search:** $f(n) = g(n)$
- Greedy: $f(n) = h(s)$
- A^* : $f(n) = h(s) + g(n)$
- WA^* : $f(n) = W * h(s) + g(n)$

Weighted A*

- $f(n) = g(n) + w * h(s)$
- If $w == 0$: $f(n) = g(n) \Rightarrow$ uniform cost
- If $w == 1$: $f(n) = g(n) + h(s) \Rightarrow A^*$
- If $w == \text{infinite}$: $f(n) = h(s) \Rightarrow$ Greedy

Properties of Heuristic functions

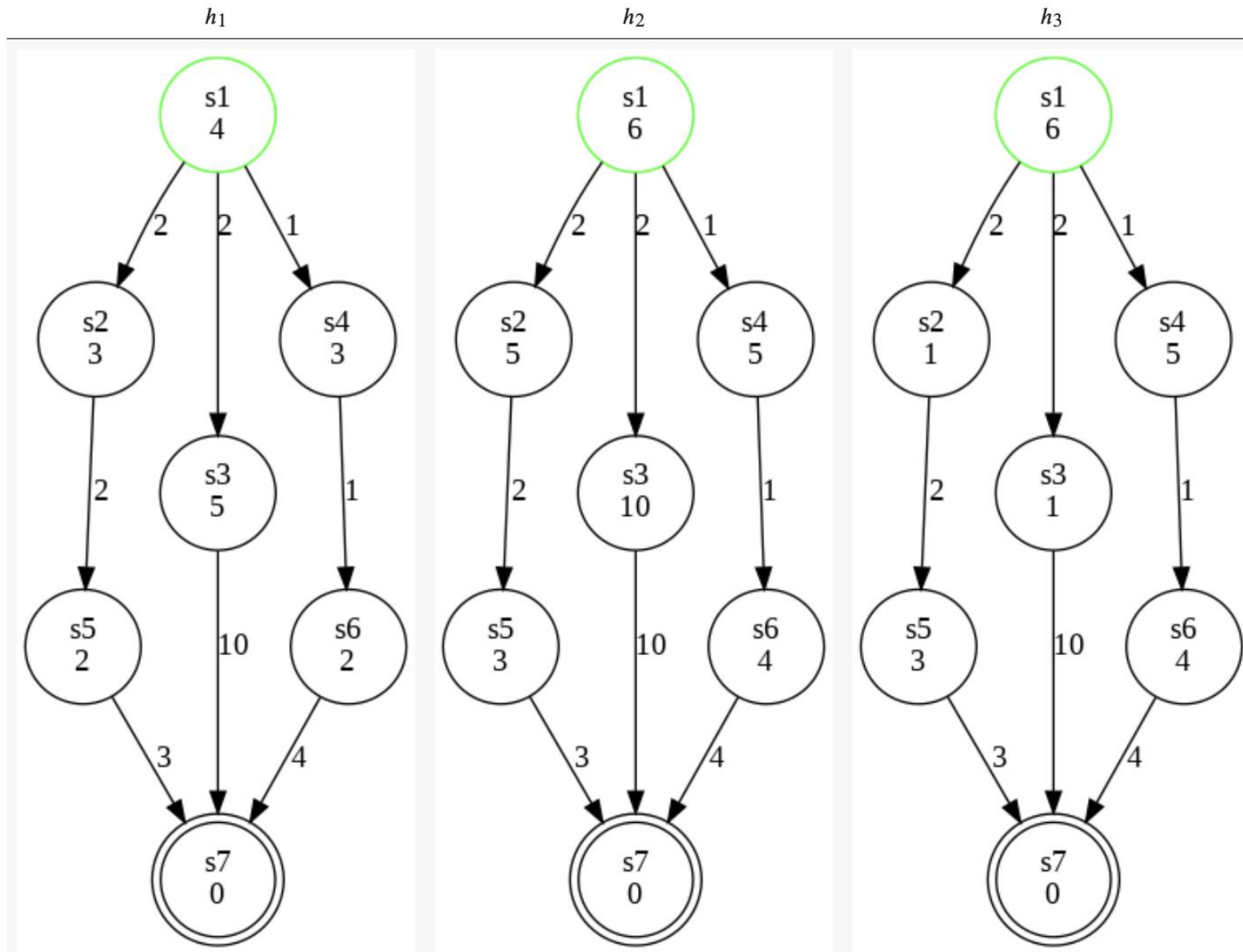
Definition (Safe/Goal-Aware/Admissible/Consistent). Let Π be a planning task with state space $\Theta_{\Pi} = (S, L, c, T, I, S^G)$, and let h be a heuristic for Π . The heuristic is called:

- **safe** if $h^*(s) = \infty$ for all $s \in S$ with $h(s) = \infty$;
- **goal-aware** if $h(s) = 0$ for all goal states $s \in S^G$;
- **admissible** if $h(s) \leq h^*(s)$ for all $s \in S$;
- **consistent** if $h(s) \leq h(s') + c(a)$ for all transitions $s \xrightarrow{a} s'$.

Dominant Relation

- If heuristic h_1 dominates heuristic h_2 :
- Then we will have $h_1(s) \geq h_2(s)$, for all s belongs to state space S
- And both h_1 and h_2 need to be admissible

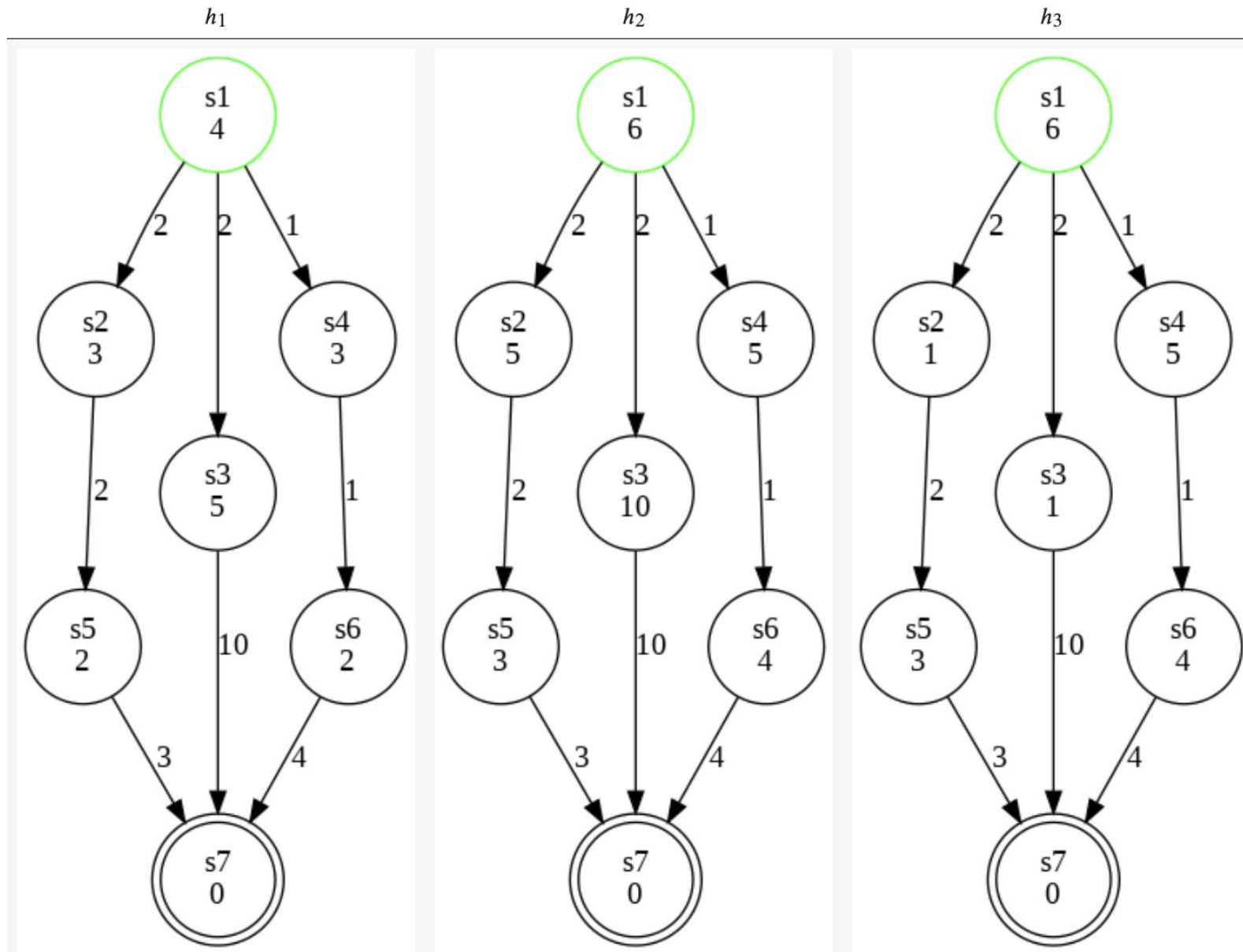
Problem 1



Task 1

- Which heuristics are admissible?
 - Which are consistent?
 - Does any of the heuristics dominate any other?
- *admissible* if $h(s) \leq h^*(s)$ for all $s \in S$;
 - *consistent* if $h(s) \leq h(s') + c(a)$ for all transitions $s \xrightarrow{a} s'$.

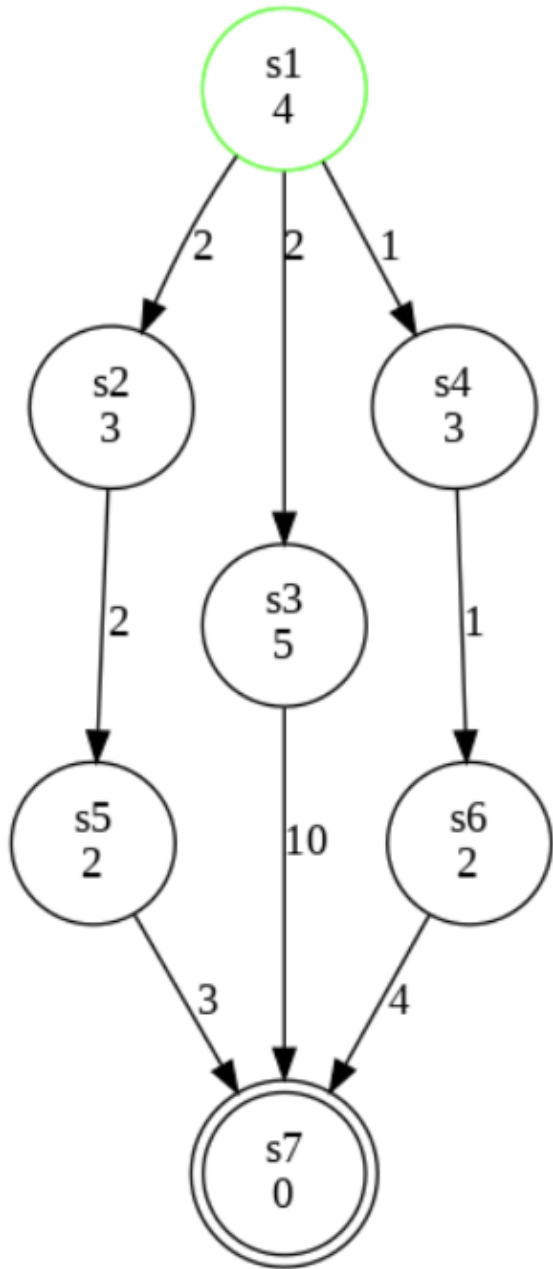
Problem 1



Task 2

- Choose one Heuristic and perform A*
- Choose one Heuristic and perform Greedy
- Choose one Heuristic and perform WA*

h_1

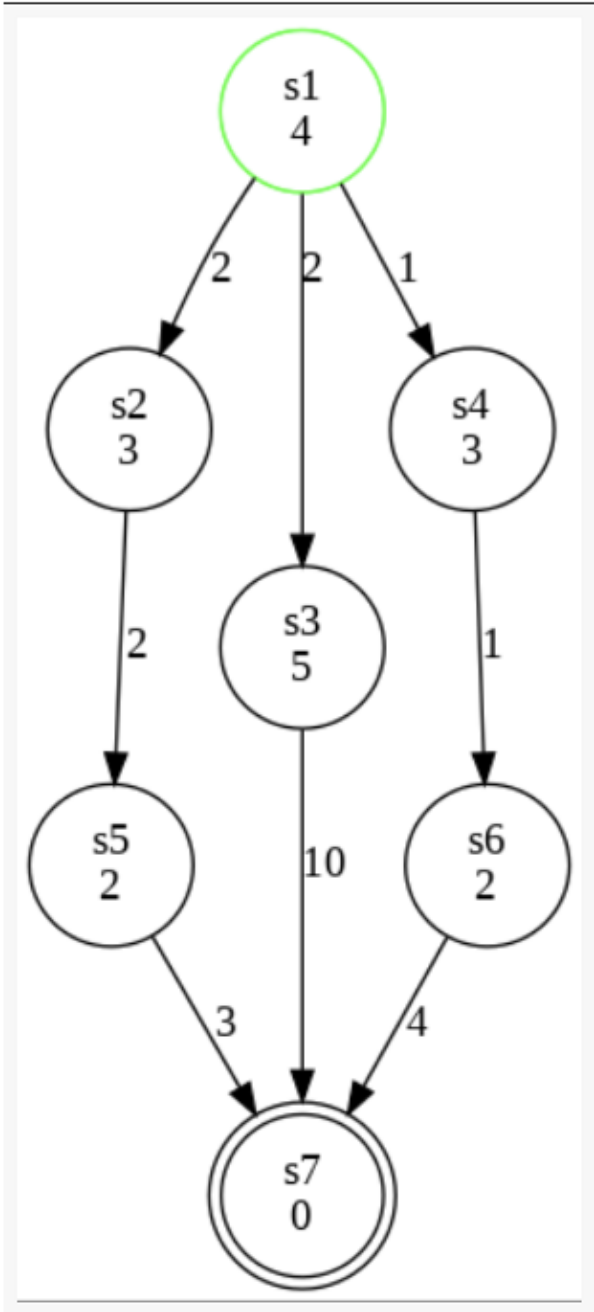


Node expansion order of A*, h_1

When pop up a node from the data structure:

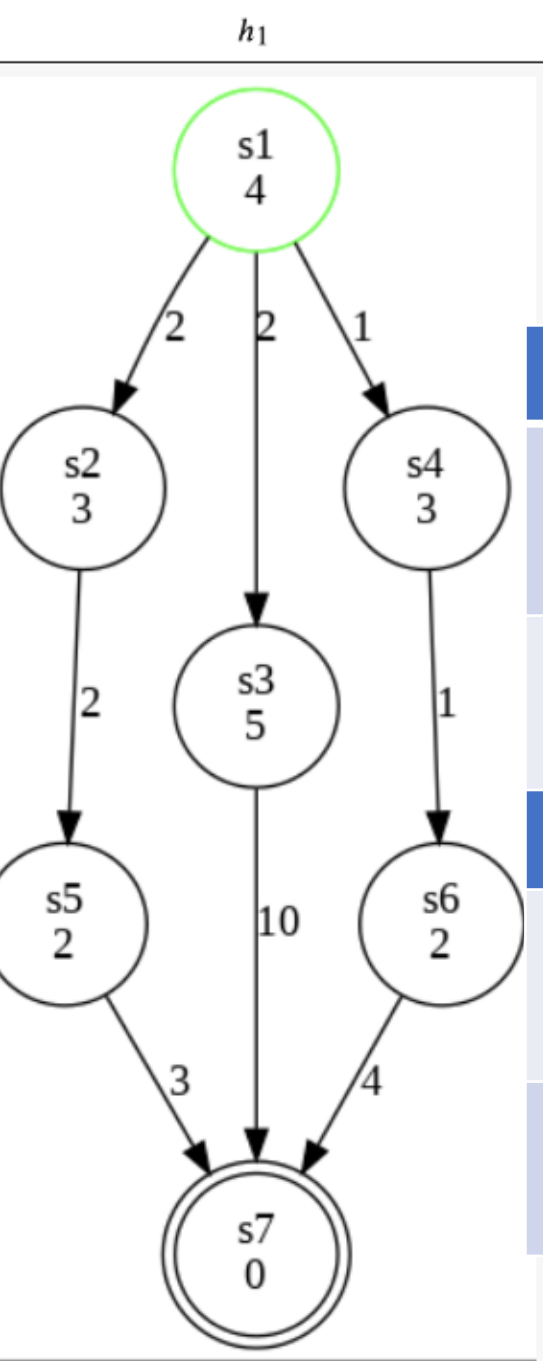
1. Check if current node n contains the goal state
2. Generate children nodes, and put into data structure

h_1



Node expansion order of A*, h_1

	I0	I1	I2
Open	$n_0 = \langle s_1, 4, 0, \text{null} \rangle$	$n_1 = \langle s_2, ?, ?, n_0 \rangle$ $n_2 = \langle s_3, ?, ?, n_0 \rangle$ $n_3 = \langle s_4, ?, ?, n_0 \rangle$	
Closed		n_0	
	I3	I4	I5
Open			
Closed			



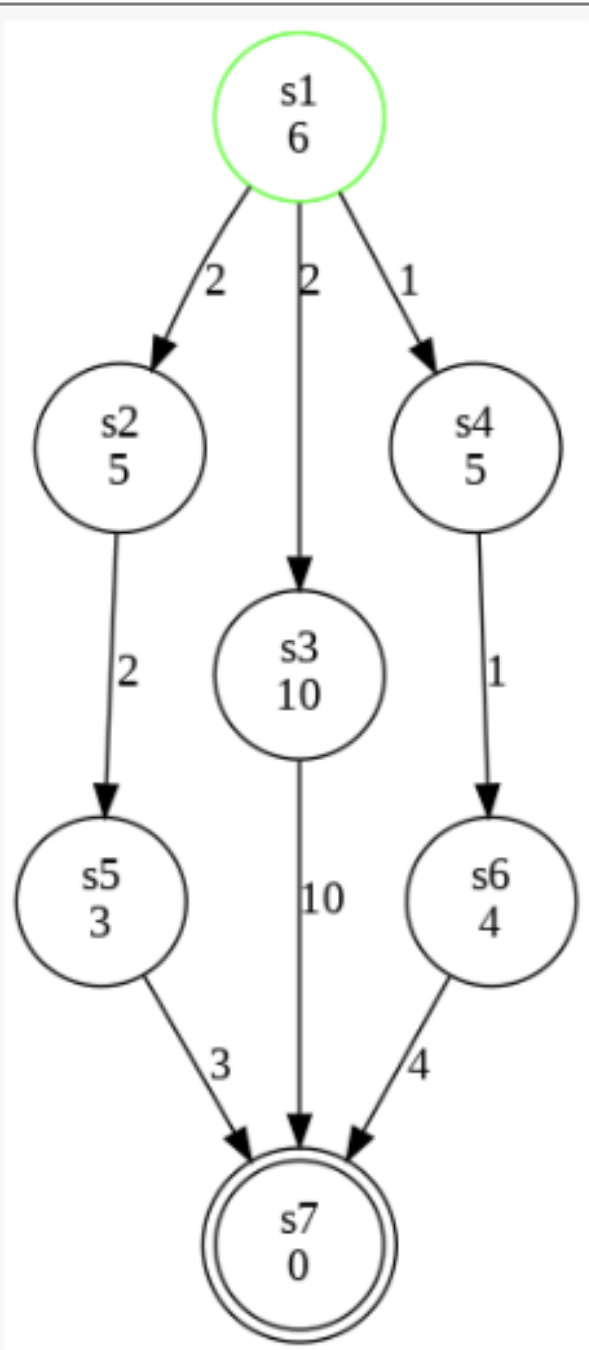
Node expansion order of A*, h_1

	I0	I1	I2
Open	$n_0 = \langle s_1, 4, 0, \text{null} \rangle$	$n_1 = \langle s_2, 5, 2, n_0 \rangle$ $n_2 = \langle s_3, 7, 2, n_0 \rangle$ $n_3 = \langle s_4, 4, 1, n_0 \rangle$	$n_1 = \langle s_2, 5, 2, n_0 \rangle$ $n_2 = \langle s_3, 7, 2, n_0 \rangle$ $n_4 = \langle s_6, 4, 2, n_3 \rangle$
Closed		n_0	n_0, n_3
	I3	I4	I5
Open	$n_1 = \langle s_2, 5, 2, n_0 \rangle$ $n_2 = \langle s_3, 7, 2, n_0 \rangle$ $n_5 = \langle s_7, 6, 6, n_4 \rangle$	$n_6 = \langle s_5, 6, 4, n_1 \rangle$ $n_2 = \langle s_3, 7, 2, n_0 \rangle$ $n_5 = \langle s_7, 6, 6, n_4 \rangle$	
Closed	n_0, n_3, n_4	n_0, n_1, n_3, n_4	n_0, n_1, n_3, n_4, n_5

Task 2

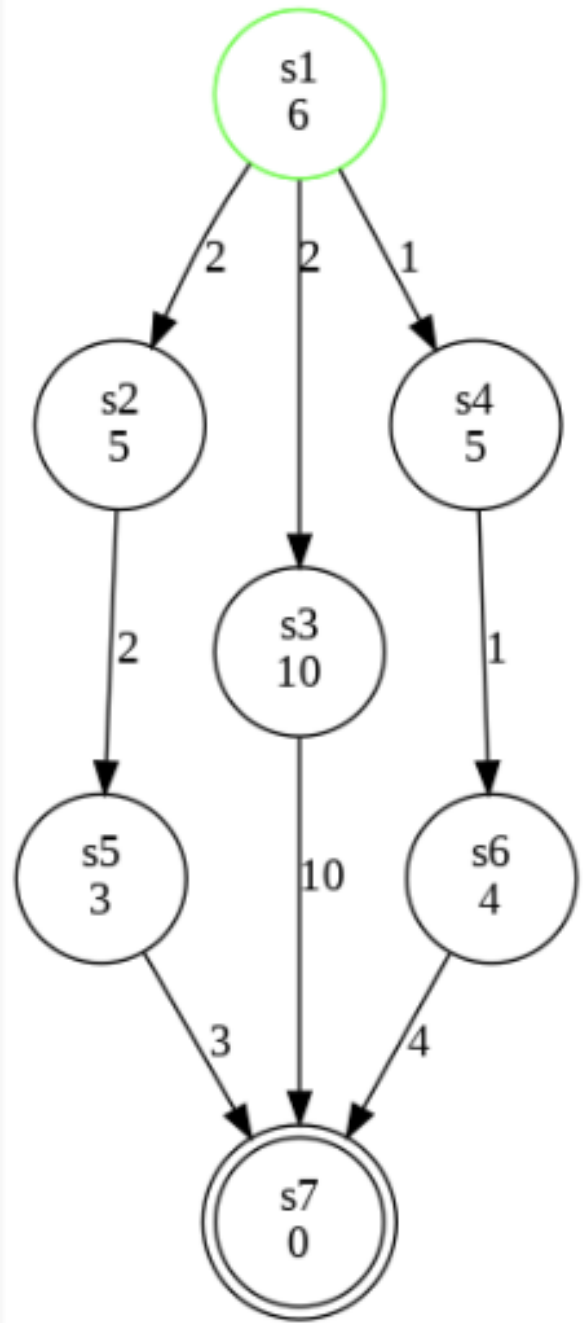
- Choose h_2 and perform A^*
- Choose h_2 and perform Greedy
- Choose h_2 and perform WA^* , weight =2

h_2



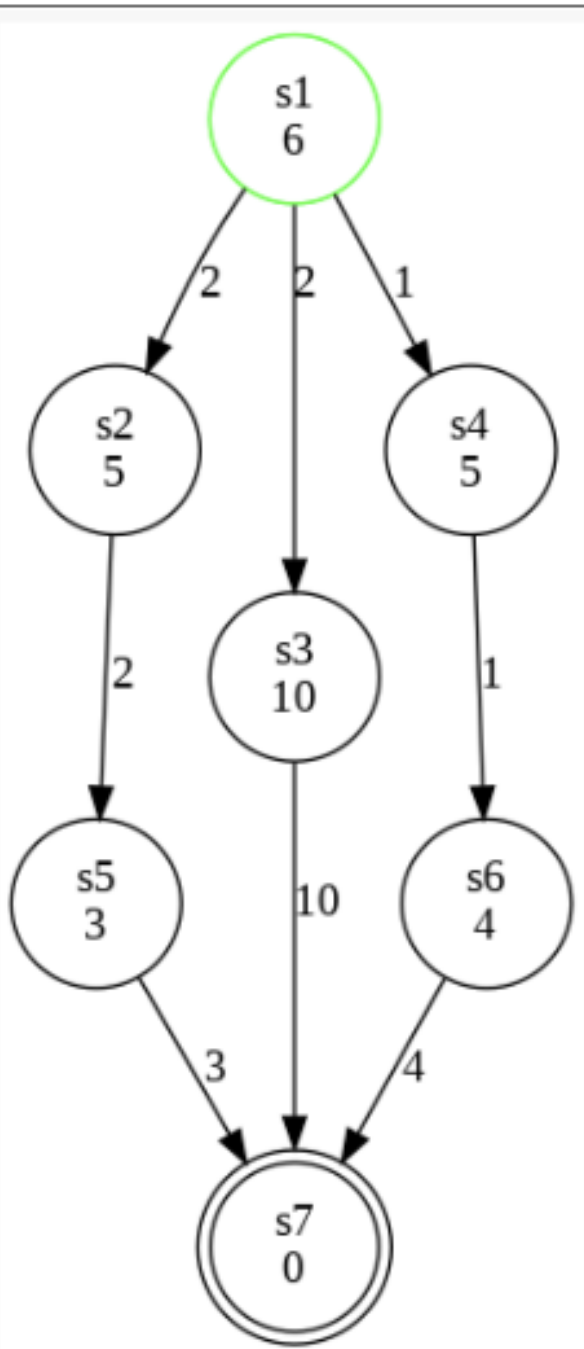
Node expansion order of A*, h_2

```
nodes = [  
    # (state, fn, accumulated cost,  
    id of parent node)  
    ('s1', 6, 0, None),  
    ('s4', 6, 1, 0),  
    ('s6', 6, 2, 1),  
    ('s7', 6, 6, 2)  
]
```

h_2 

Node expansion order of Greedy, h_2

```
nodes = [  
    # (state, fn, accumulated cost, id  
    of parent node)  
    ('s1', 6, 0, None),  
    ('s4', 5, 1, 0),  
    ('s6', 4, 2, 1),  
    ('s7', 0, 6, 2)  
]
```

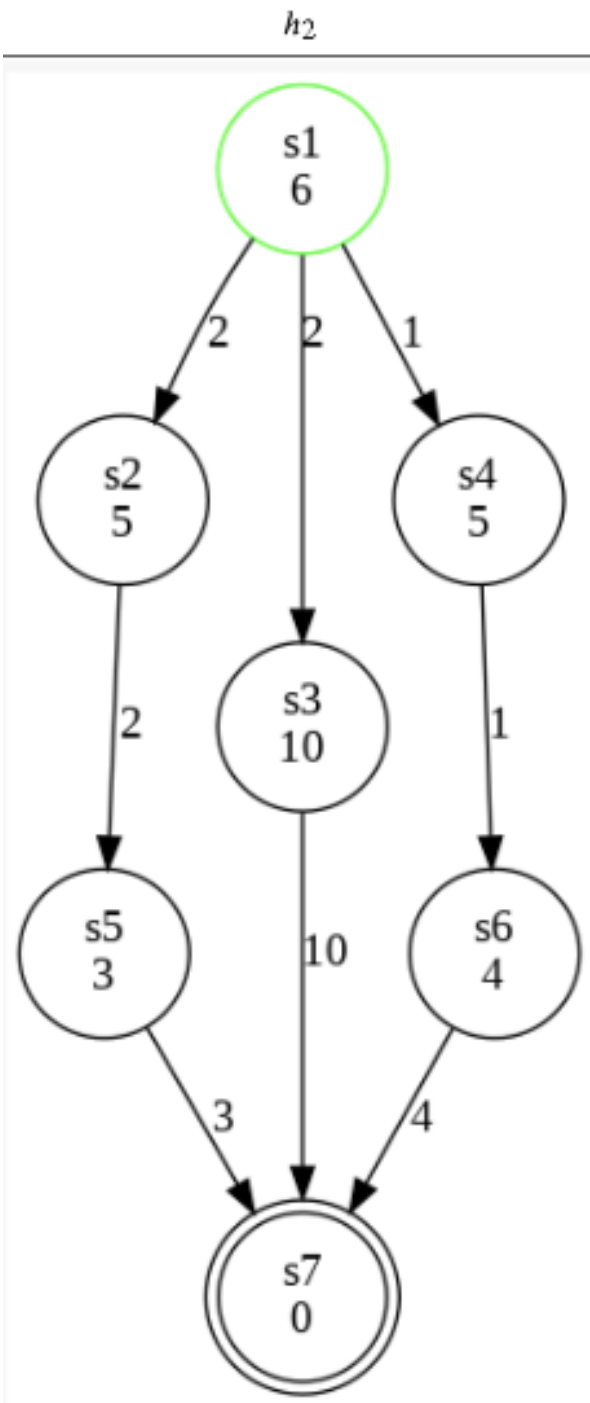

h_2 

Node expansion order of WA*, h_2

```
nodes = [  
    # (state, fn, accumulated cost,  
    id of parent node)  
    ('s1', 12, 0, None),  
    ('s4', 11, 1, 0),  
    ('s6', 10, 2, 1),  
    ('s7', 6, 6, 2)  
]
```

Task 2

- Which is the path returned as the solution?
 - Is this the optimal plan?
- (using h_2 and A^* as example)



Problem 2

- Consider an $m \times m$ **Manhattan Grid**, and a set of coordinates G to visit in any order.
- **Hint:** Consider a set of coordinates V' remaining to be visited, or a set of coordinates V already visited. What's the difference between them
- Formulate a state-based search problem to find a tour of all the desired points (i.e. define a state space, applicable actions, transition and cost functions).
- What is the branching factor of this search?
- What is the size of the state space in terms of m and G ?

m x m Grid

		Assume We are here		

$P = \{S, s0, SG, A, T, C\}$

Consider 2 ways :

- Using a set of coordinates V' remaining to be visited,
- Or a set of coordinates V already visited.
- What's the difference between them

Problem 2

- $S = \{\langle x, y, V' \rangle \mid x, y \in \{0, \dots, m-1\} \wedge V' \subseteq G\}$
- $s_0 = \langle (0, 0), G \setminus \{(0, 0)\} \rangle$
- $S_G = \{\langle (x, y), \{\} \rangle \mid x, y \in \{0, \dots, m-1\}\}$
- $A(\langle x, y, V' \rangle) = \{(dx, dy) \mid$
 - $dx, dy \in \{-1, 0, 1\}$
 - $\wedge |dx| + |dy| = 1$
 - $\wedge x + dx, y + dy \in \{0, \dots, m-1\}$
 - $(x + dx, y + dy) \notin W \}$
- $T(\langle x, y, V' \rangle, (dx, dy)) = \langle x + dx, y + dy, V' \setminus \{(x + dx, y + dy)\} \rangle$
- $c(a, s) = 1$

m x m Grid

		Assume We are here		

$P = \{S, s0, SG, A, T, C\}$

Consider 2 ways :

- Using a set of coordinates V' remaining to be visited,
- Or a set of coordinates V already visited.
- What's the difference between them

$m \times m$ Grid



- What is the branching factor of this search?
- What is the size of the state space in terms of m and G ?

m x m Grid

		Assume We are here		

- What is the branching factor of this search?
- What is the size of the state space in terms of m and G ?

If using V' , then $m^2 \times 2^{|G|}$

If using V , then $m^2 \times 2^{|m \times m|}$