Week 4: STRIPS and Heuristic

COMP90054 – AI Planning for Autonomy

Key concepts

- STRIPS problem
- Heuristic functions

Problem 1

Consider a m x m manhattan grid, and a set of coordinates G′ to visit in any order, and a set of inaccessible coordinates (walls) W

a state = <current coordinate, a set of remaining coordinates>

State-space model

P = <S,s_0, S_G, A, T, c>

S = State space

s_0 = initial state

S_G = goal states

A = actions

T = transition functions

c = costs

Initial state $\boldsymbol{s_0} = <(\mathbf{0},\mathbf{0}), \boldsymbol{G'}\backslash\{(\mathbf{0},\mathbf{0})\}>$

Goal state $\boldsymbol{S_G} = \{<(\boldsymbol{x},\boldsymbol{y}),\{\}> \mid \boldsymbol{x},\boldsymbol{y} \in \{\mathbf{0},\dots,\boldsymbol{m}-\mathbf{1}\}\}$

State $\boldsymbol{S} = \{<(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{V'}> \mid \boldsymbol{x},\boldsymbol{y} \in \{\mathbf{0},\dots,\boldsymbol{m}-\mathbf{1}\} \wedge \boldsymbol{V'} \subseteq \boldsymbol{G'}\}$

Action A($<(\boldsymbol{x},\boldsymbol{y}), \boldsymbol{V'}>$**)** $= \{(\boldsymbol{dx},\boldsymbol{dy}) \mid \boldsymbol{dx},\boldsymbol{dy} \in \{-\mathbf{1},\mathbf{0},\mathbf{1}\}$
$\wedge\ |\boldsymbol{dx}| + |\boldsymbol{dy}| = \mathbf{1}$
$\wedge\ \boldsymbol{x}+\boldsymbol{dx}, \boldsymbol{y}+\boldsymbol{dy} \in \{\mathbf{0},\dots,\boldsymbol{m}-\mathbf{1}\}$
$\wedge\ (\boldsymbol{x}+\boldsymbol{dx}, \boldsymbol{y}+\boldsymbol{dy}) \notin \boldsymbol{W}\}$

Transition f($<(\boldsymbol{x},\boldsymbol{y}), \boldsymbol{V'}>, (\boldsymbol{dx},\boldsymbol{dy})$**)** $= <(\boldsymbol{x}+\boldsymbol{dx}, \boldsymbol{y}+\boldsymbol{dy}),\ \boldsymbol{V'}\backslash\{(\boldsymbol{x}+\boldsymbol{dx}, \boldsymbol{y}+\boldsymbol{dy})\}>$

Cost c($\boldsymbol{a}$**) = 1**

Problem 1

Consider a m x m manhattan grid, and a set of coordinates G′ to visit in any order, and a set of inaccessible coordinates (walls) W

- A **problem** in STRIPS is a tuple $P = \langle F, O, I, G \rangle$:
  - F stands for set of all atoms (boolean vars) (predicates, facts, fluents)
  - O stands for set of all operators (actions)
  - $I \subseteq F$ stands for initial situation
  - $G \subseteq F$ stands for goal situation
- Operators $o \in O$ **represented** by
  - the Add list $Add(o) \subseteq F$ True
  - the Delete list $Del(o) \subseteq F$ False
  - the Precondition list $Pre(o) \subseteq F$

$$\boldsymbol{I} \quad = \{\boldsymbol{at(0,0), visited(0,0)}\}$$

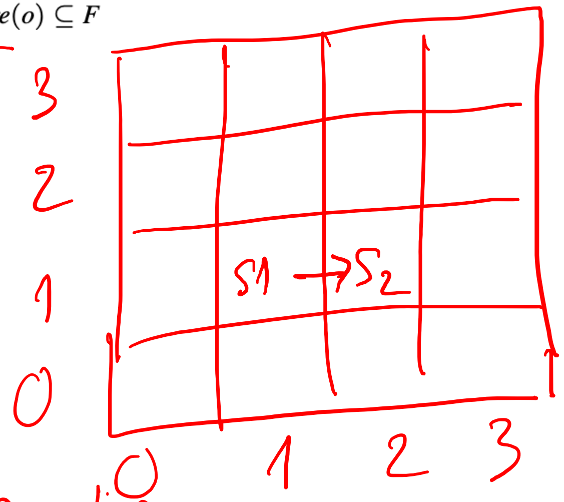
$$\boldsymbol{G} \quad = \{\boldsymbol{visited(x,y) | x, y \in G'}\}$$

$$\boldsymbol{F} \quad = \{\boldsymbol{at(x,y), visited(x,y) | x, y \in \{0, \dots, m-1\}}\}$$

$(x,y) \rightarrow (x',y')$

$\underset{\text{False}}{at(x,y)} \rightarrow \underset{\text{True}}{at(x',y')}, \underset{\text{True}}{visited(x',y')}$

O = {$\boldsymbol{move(x,y,x',y')}$:
- *Prec: at(x,y)*
- *Add: at(x′, y′), visited(x′,y′)*
- *Del: at(x, y) | for each adjacent (x,y),(x′,y′), and (x′,y′) ∉ W* }



pre: at(1,1)

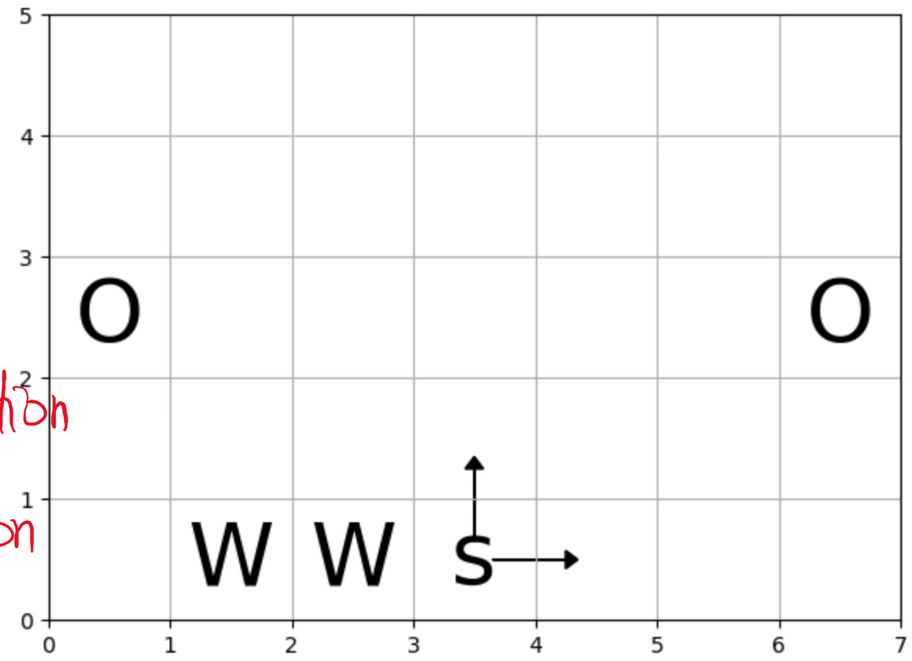
$S1 : (1,1) \xrightarrow{\text{action}} S_2 : (2,1)$

Problem 2

1. Zero heuristic
2. Goal-counting heuristic
3. Manhattan Distance to Closest Goal heuristic
4. Manhattan Distance to Furthest Goal heuristic
5. Sum of manhattan distances of all goals
6. Minimum spanning tree

1) Bad heuristic: ↑ search time + ↓ heuristic calculation

2) Good heuristic: ↓ search time + ↑ heuristic calculation



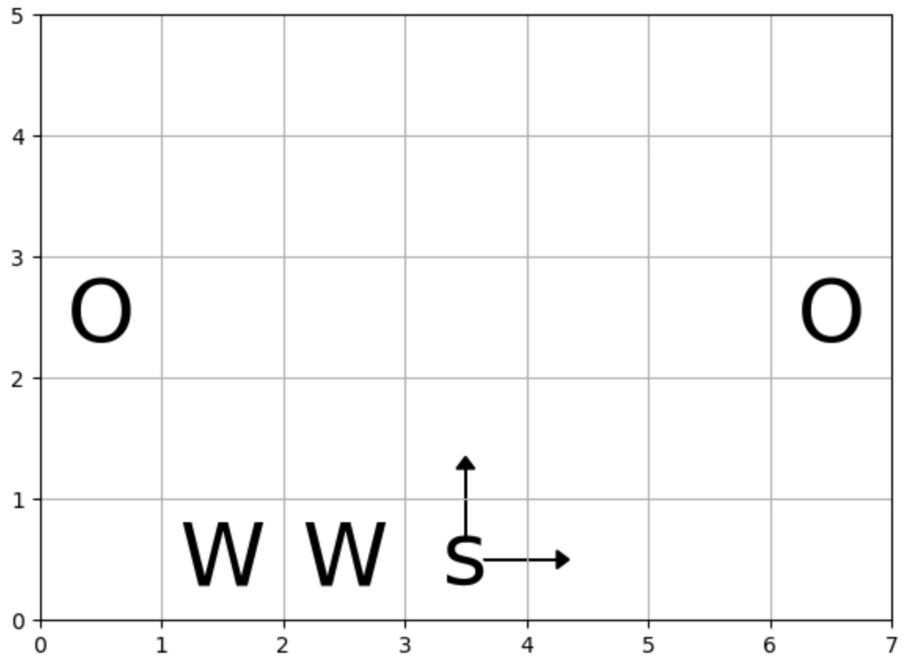
$\underbrace{\text{Number of node expansion}}_{\text{Search time}}$ + Calculation time of the heuristic function = Total running time

Tip: If h1 dominates h2 then A with h1 will expand less or equal node to h2*

Problem 2

1. Zero heuristic h = 0

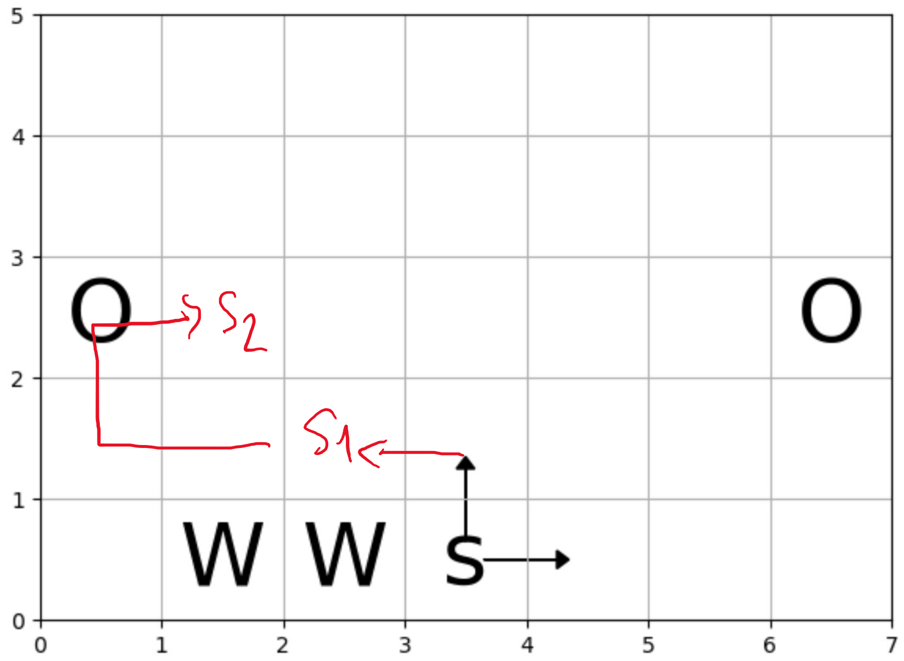
Admissible: Yes
Consistent: Yes
Time to calculate h: None



Problem 2

2. Goal-counting heuristic

Admissible: Yes
Consistent: Yes
Time to calculate h: Easy



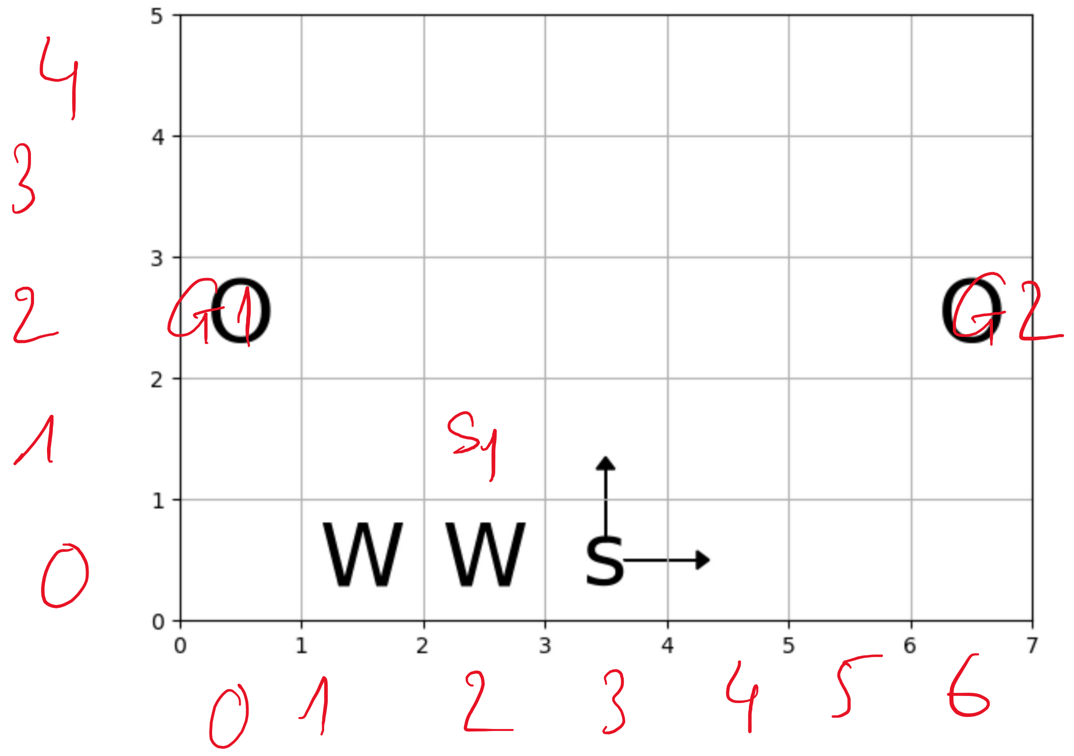
$h(s_1) = 2$

$h(s_2) = 1$

Problem 2

3. Manhattan Distance to Closest Goal heuristic

Admissible: Yes
Consistent: Yes
Time to calculate h: Easy

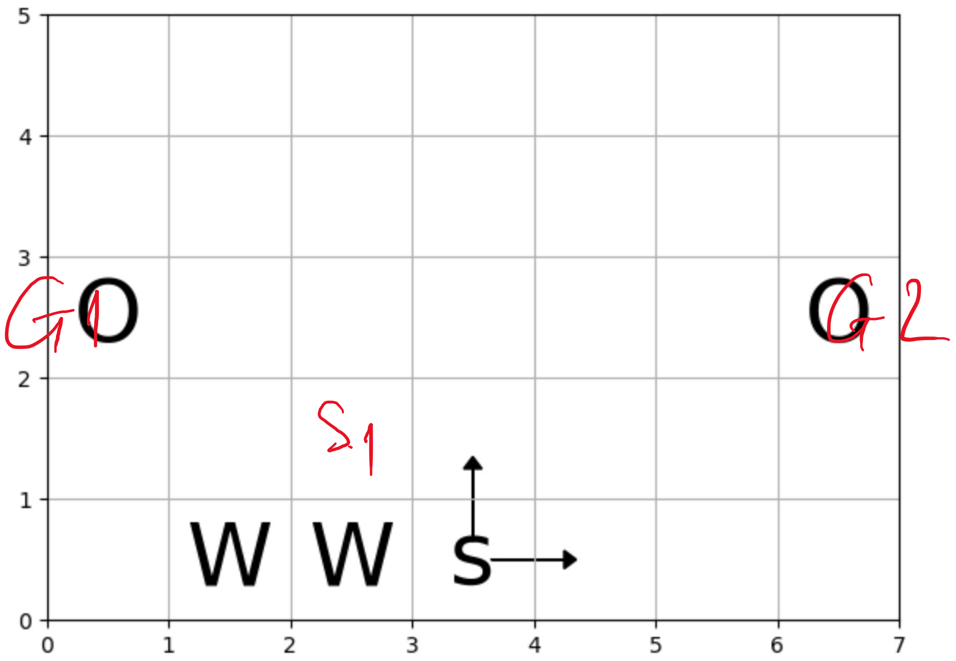


$$h(s_1) = \min(d(s_1, G1), d(s_1, G2))$$
$$= \min(3, 5)$$
$$= 3$$

Problem 2

4. Manhattan Distance to Furthest Goal heuristic

Admissible: Yes
Consistent: Yes
Time to calculate h: Easy

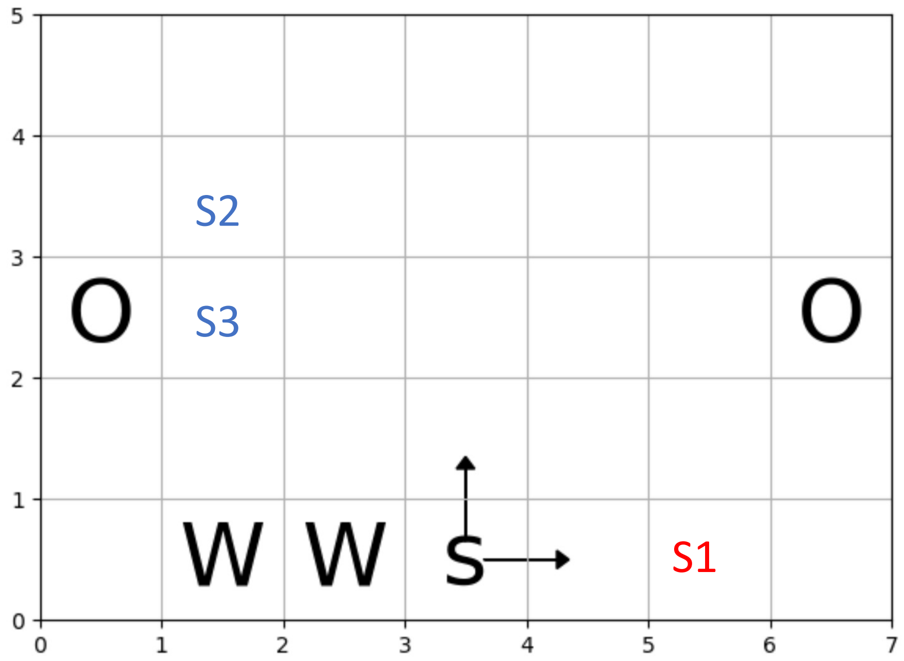


$$h(s_1) = \max\left(d(s_1, G_1), d(s_1, G2)\right)$$
$$= \max(3, 5)$$
$$= 5$$

Problem 2

5. Sum of Manhattan distances of all goals

Admissible: No
Consistent: No
Time to calculate h: Easy



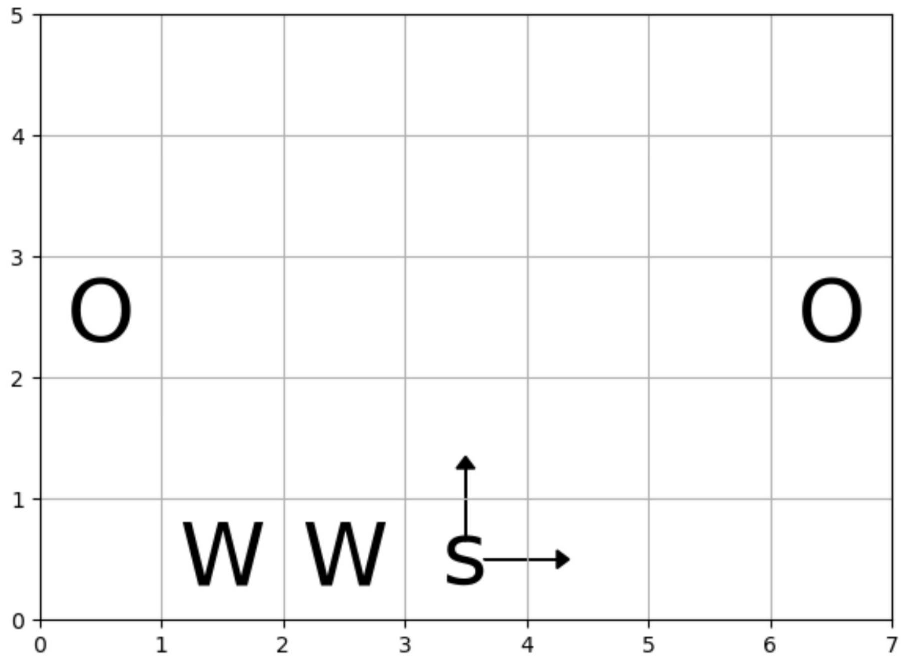
h(s1) = 3 + 7 = 10
h*(s1) = 3 + 6 = 9
h(s1) > h*(s1) => Not admissible

h(s2) = 2 + 6 = 8
h(s3) = 1 + 5 = 6
h(s2) – h(s3) = 2
c(s2, s3) = 1
=> h(s2) – h(s3) > c(s2, s3) => Not consistent

Problem 2

Dominate relation

1. Zero heuristic: Admissible, Consistent
2. Goal-counting heuristic: Admissible, Consistent
3. Manhattan Distance to Closest Goal heuristic: Admissible, Consistent
4. Manhattan Distance to Furthest Goal heuristic: Admissible, Consistent
5. Sum of manhattan distances of all goals: Not admissible, Not consistent



h(goal counting) > h(zero)
h(closest) > h(zero)
h(furthest) > h(zero)

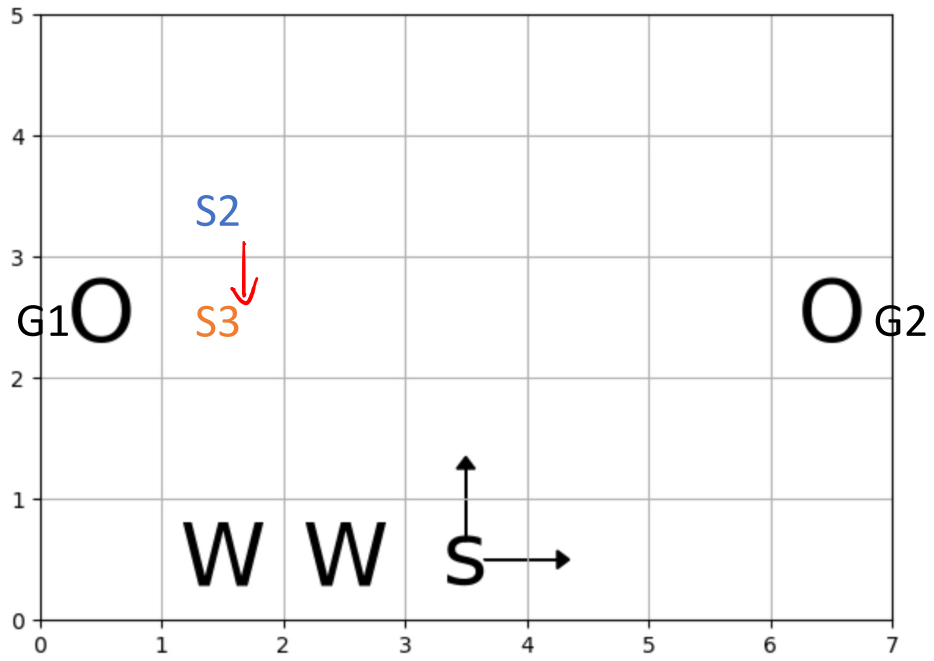
h(furthest) > h(closest)
h(furthest) > h(goal counting)

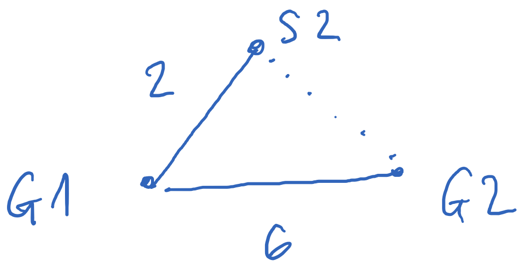
Problem 2

6. Minimum spanning tree

Admissible: Yes
Consistent: No
Time to calculate h: Medium

Minimum spanning tree: Only select a subset of edges to connect all vertices and the total cost is minimum

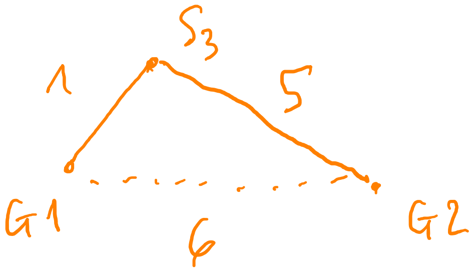




$$h(S_2) = 2 + 6 = 8$$

$$\Rightarrow h(S_2) - h(S_3) = 8 - 6 = 2$$

$$> c(S_2, S_3) = 1$$



$$h(S_3) = 1 + 5 = 6$$