CS3243 Tutorial 3

Heuristics

Gu Zhenhao

February 8, 2023

Admissibility & Consistency

How to judge if a heuristic is good?

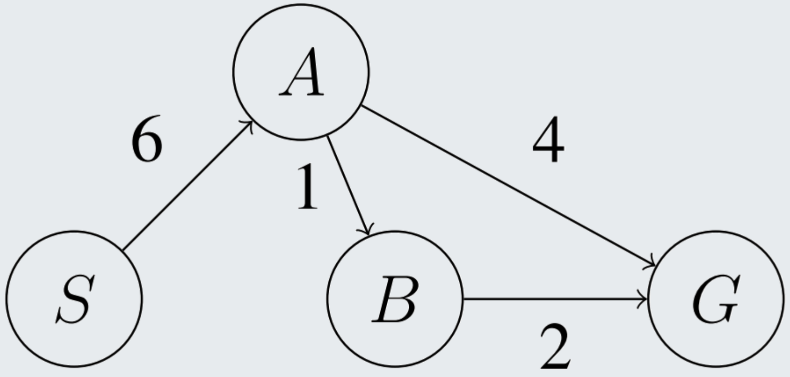
Summary

- **admissible** (optimistic), $h(n) \leq h^*(n)$ for all n,
  - **Property**: for all states n on the path to goal G, $f(n) \leq f(G)$.
  - Tree search and Graph search (version 2) are optimal.

- **consistent**, $h(n) \leq c(n, a, n') + h(n')$ for all n and its successor n'.
  - **Property**: f value never decreases along an optimal path to a goal.
  - Graph search (version 2 & 3) is optimal*.

** Needs both consistency and admissibility.*

Tutorial 2, Problem 5.a

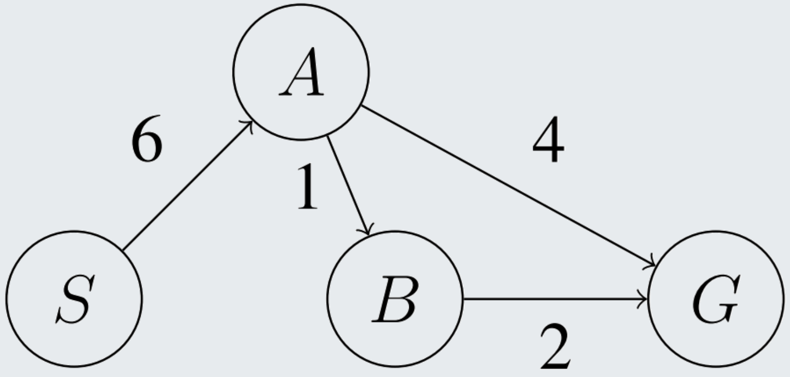


- **admissibility**, $h(n) \leq h^*(n)$ for all n.
- **consistency**, $h(n) \leq c(n, a, n') + h(n')$.

	S	A	B	G	**Admissible?**	**Consistent?**
h_1	0	0	0	0		
h_2	8	1	1	0		
h_3	9	3	2	0		
h_4	6	3	1	0		
h_5	8	4	2	0		

$\boldsymbol{h^*}$				

Tutorial 2, Problem 5.a

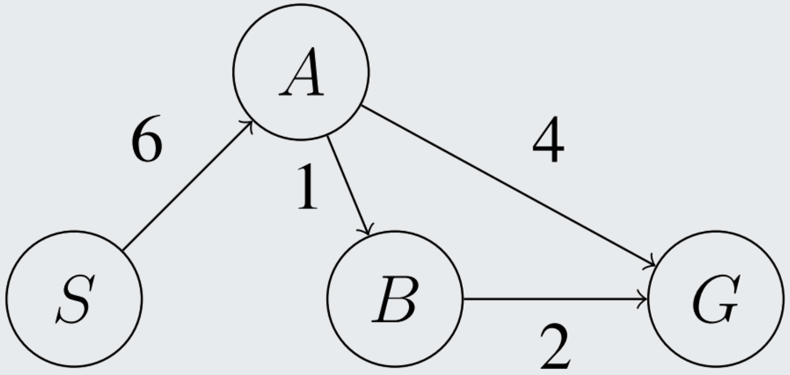


- **admissibility**, $h(n) \leq h^*(n)$ for all n.
- **consistency**, $h(n) \leq c(n, a, n') + h(n')$.

	S	A	B	G	**Admissible?**	**Consistent?**
h_1	0	0	0	0	Yes	
h_2	8	1	1	0	Yes	
h_3	9	3	2	0	Yes	
h_4	6	3	1	0	Yes	
h_5	8	4	2	0	No	

h^*	**9**	**3**	**2**	**0**

Tutorial 2, Problem 5.a

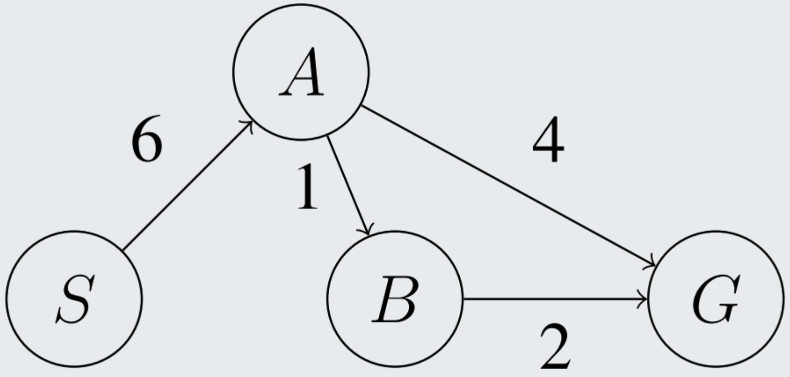


- **admissibility**, $h(n) \leq h^*(n)$ for all n.
- **consistency**, $h(n) \leq c(n, a, n') + h(n')$.

	S	A	B	G	**Admissible?**	**Consistent?**
h_1	0	0	0	0	Yes	Yes
h_2	8	1	1	0	Yes	
h_3	9	3	2	0	Yes	
h_4	6	3	1	0	Yes	
h_5	8	4	2	0	No	

h^*	**9**	**3**	**2**	**0**

Tutorial 2, Problem 5.a



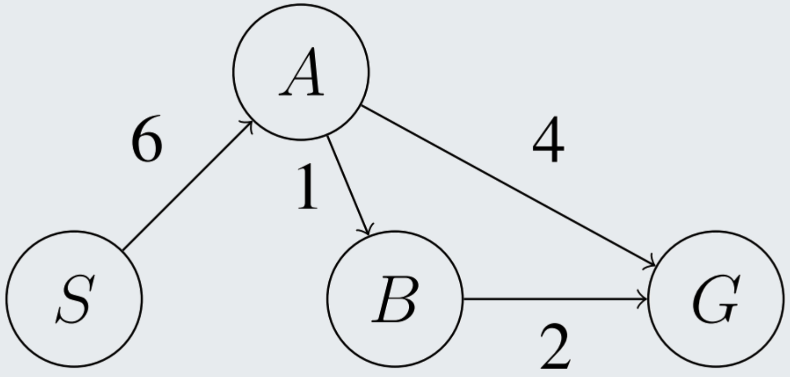
- **admissibility**, $h(n) \leq h^*(n)$ for all n.
- **consistency**, $h(n) \leq c(n, a, n') + h(n')$.

	S	A	B	G	**Admissible?**	**Consistent?**
h_1	0	[illegible]	[illegible]	0	Yes	Yes
h_2	8	1	1	0	Yes	No
h_3	9	3	2	0	Yes	
h_4	6	3	1	0	Yes	
h_5	8	4	2	0	No	

$h(S) = 8 > h(A) + c(S, A) = 7$

h^*	**9**	**3**	**2**	**0**

Tutorial 2, Problem 5.a

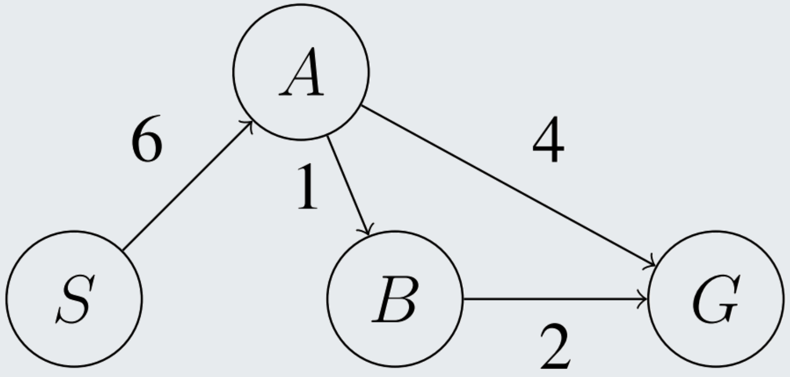


- **admissibility**, $h(n) \leq h^*(n)$ for all n.
- **consistency**, $h(n) \leq c(n, a, n') + h(n')$.

	S	A	B	G	**Admissible?**	**Consistent?**
h_1	0	0	0	0	Yes	Yes
h_2	8	1	1	0	Yes	No
h_3	9	3	2	0	Yes	Yes
h_4	6	3	1	0	Yes	
h_5	8	4	2	0	No	

h^*	**9**	**3**	**2**	**0**

Tutorial 2, Problem 5.a



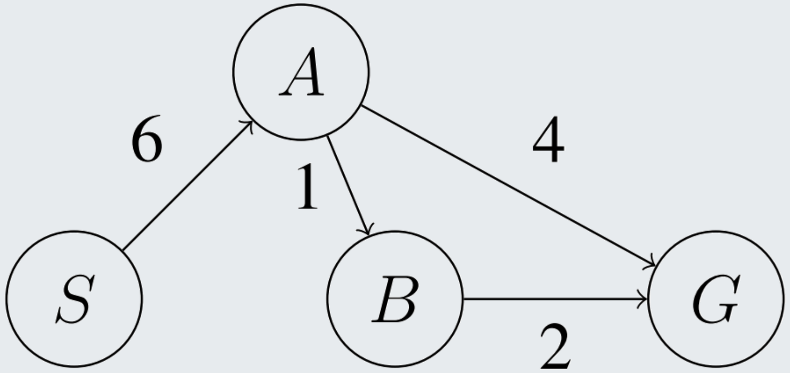
- **admissibility**, $h(n) \leq h^*(n)$ for all n.
- **consistency**, $h(n) \leq c(n, a, n') + h(n')$.

	S	A	B	G	**Admissible?**	**Consistent?**
h_1	0	0	0	0	Yes	Yes
h_2	8	1	1	0	Yes	No
h_3	9	3			Yes	Yes
h_4	6	3	1	0	Yes	No
h_5	8	4	2	0	No	

$h(A) = 3 > h(B) + c(A, B) = 2$

h^*	**9**	**3**	**2**	**0**

Tutorial 2, Problem 5.a



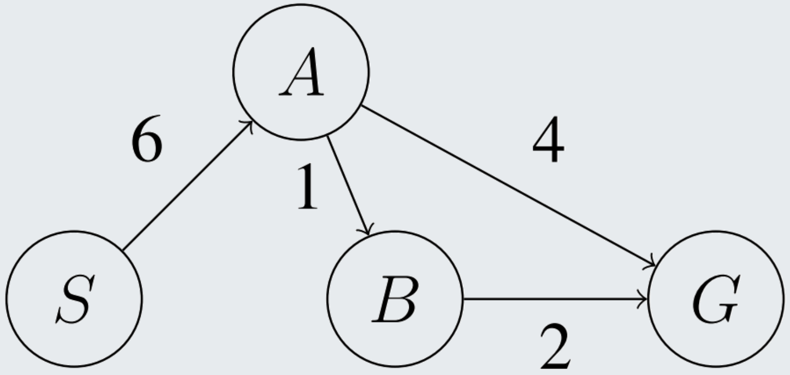
- **admissibility**, $h(n) \leq h^*(n)$ for all n.
- **consistency**, $h(n) \leq c(n, a, n') + h(n')$.

	S	A	B	G	**Admissible?**	**Consistent?**
h_1	0	0	0	0	Yes	Yes
h_2	8	1	1	0	Yes	No
h_3	9	3	2	0	Yes	Yes
h_4	6	3			Yes	No
h_5	8	4	2	0	No	No

$h(A) = 4 > h(B) + c(A, B) = 3$

h^*	9	3	2	0

Tutorial 2, Problem 5.b



h_3 is the best as $h_3 = h^*$,

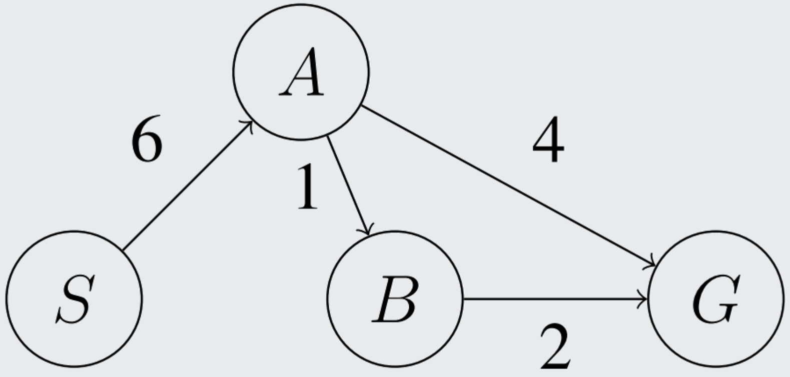
$\Rightarrow$ it is admissible and consistent,

$\Rightarrow$ dominates other admissible heuristics.

	S	A	B	G	**Admissible?**	**Consistent?**
h_1	0	0	0	0	Yes	Yes
h_2	8	1	1	0	Yes	No
h_3	9	3	2	0	Yes	Yes
h_4	6	3	1	0	Yes	No
h_5	8	4	2	0	No	No

$\boldsymbol{h^*}$	**9**	**3**	**2**	**0**

Tutorial 2, Problem 5.c



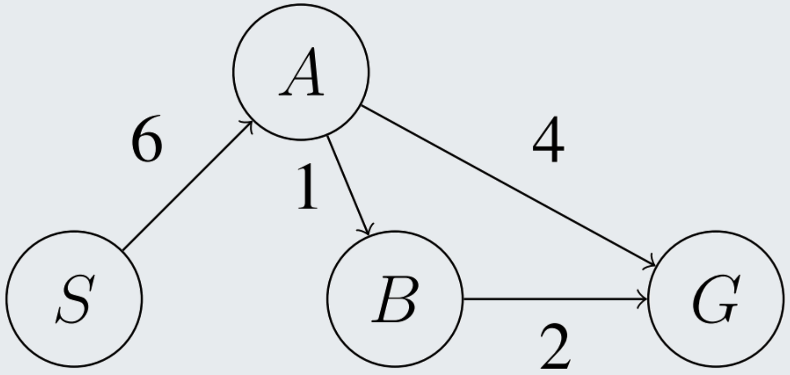
Question: is $\max\{h_3(n), h_5(n)\}$ admissible?

No! $\max\{h_3(A), h_5(A)\} = 4 > h^*(A)$.

	S	A	B	G	**Admissible?**	**Consistent?**
h_1	0	0	0	0	Yes	Yes
h_2	8	1	1	0	Yes	No
h_3	9	3	2	0	Yes	Yes
h_4	6	3	1	0	Yes	No
h_5	8	4	2	0	No	No

h^*	**9**	**3**	**2**	**0**

Tutorial 2, Problem 5.b



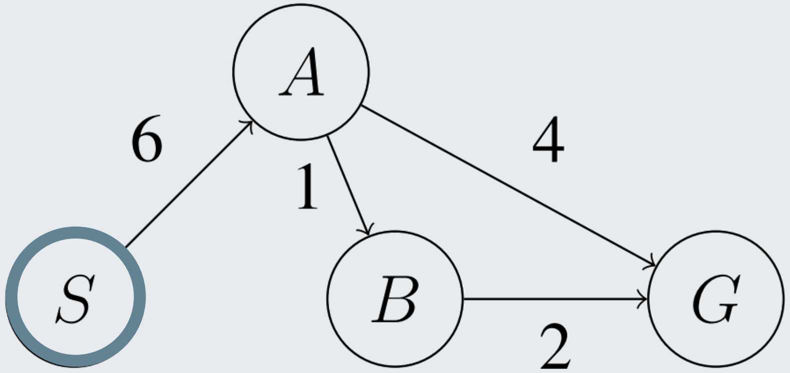
	S	A	B	G
h_4	**6**	**3**	**1**	**0**

A* graph search using h_4.

Frontier: [S(/), 0+**6**]

Visited nodes:

Tutorial 2, Problem 5.b



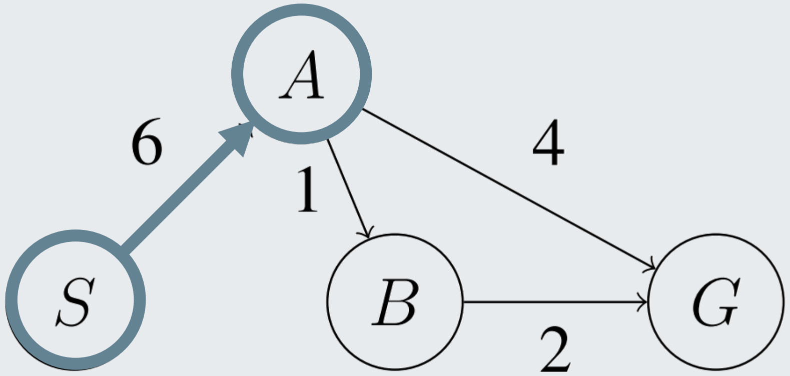
	S	A	B	G
h_4	6	3	1	0

A* graph search using h_4.

Frontier: [$A(S)$, 6+**3**]

Visited nodes: $S(/)$,

Tutorial 2, Problem 5.b



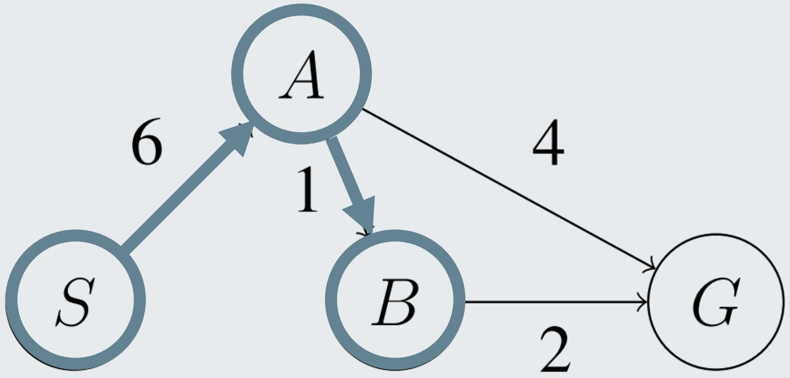
	S	A	B	G
h_4	6	3	1	0

A* graph search using h_4.

Frontier: [$B(A)$, 7+**1**], [$G(A)$, 10+**0**]

Visited nodes: $S(/)$, $A(S)$,

Tutorial 2, Problem 5.b



	S	A	B	G
h_4	6	3	1	0

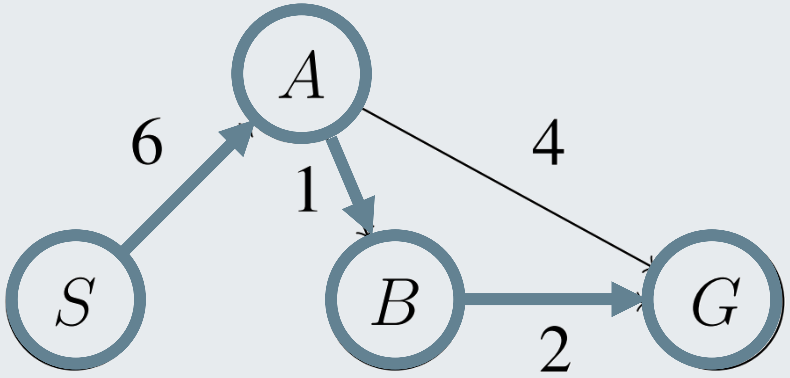
Can push G again because it hasn't been popped from frontier yet.

A* graph search using h_4.

Frontier: [$G(A)$, 10+**0**], [$G(B)$, 9+**0**]

Visited nodes: $S(/)$, $A(S)$, $B(A)$

Tutorial 2, Problem 5.b



	S	A	B	G
h_4	6	3	1	0

A* graph search using h_4.

Frontier: [$G(A)$, 10+**0**]

Visited nodes: $S(/)$, $A(S)$, $B(A)$, $G(B)$

Building Heuristics

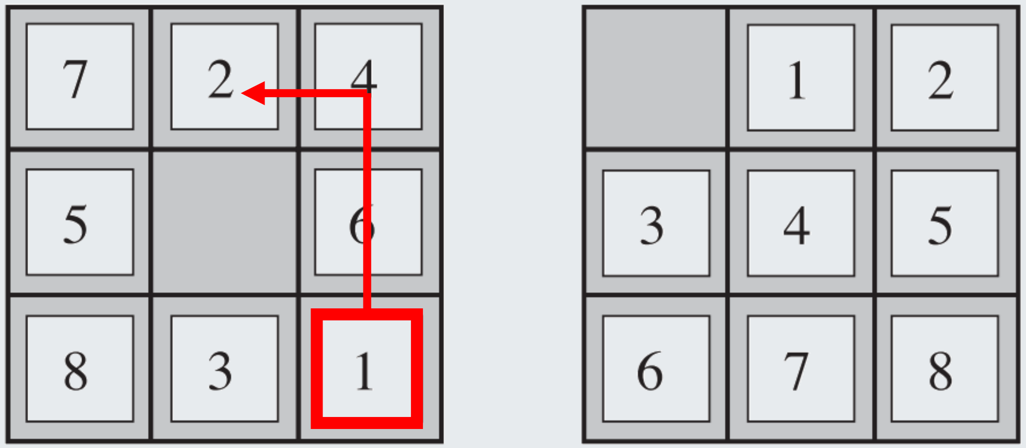
How to define a good heuristic function?

Strategies

Idea 1: Find the ***optimal cost*** of a relaxed (simplified) problem.

Strategies:

- Relax the rules.



Start State   Goal State

e.g. a tile can only be moved to a position that is

- ~~empty, and~~
- adjacent to it

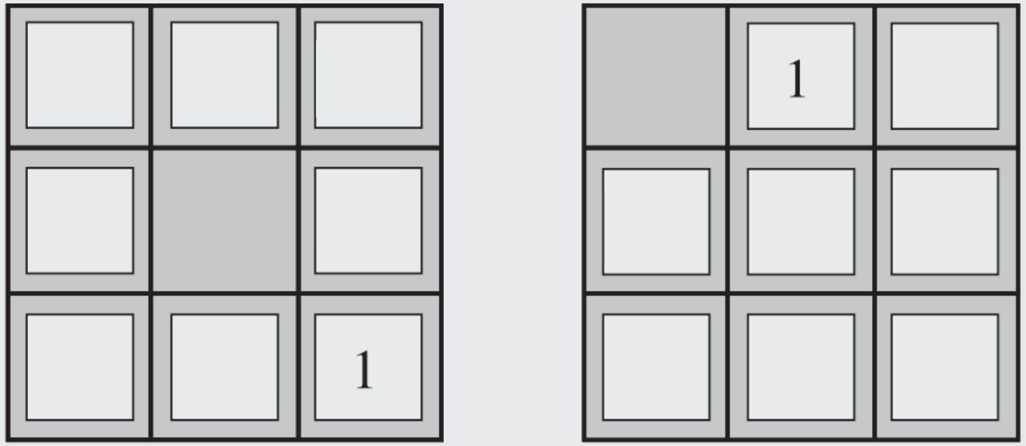
Optimal cost $=$ sum of Manhattan distances.

Strategies

Idea 1: Find the *optimal cost* of a relaxed (simplified) problem.

Strategies:

- Relax the rules.
- Solve a sub-problem.



Start State Goal State

only tile 1

e.g. the goal is to place ~~all tiles 1-9~~ to their correct positions.

Optimal cost = **minimum** no. of steps to move tile 1 to the correct position.

Strategies

Idea 1: Find the ***optimal cost*** of a relaxed (simplified) problem.

Strategies:

- Relax the rules.
- Solve a sub-problem.
- Learn from experience*.

** Sometimes used by learning agents, which is not in the scope of CS3243.*

Problem 1

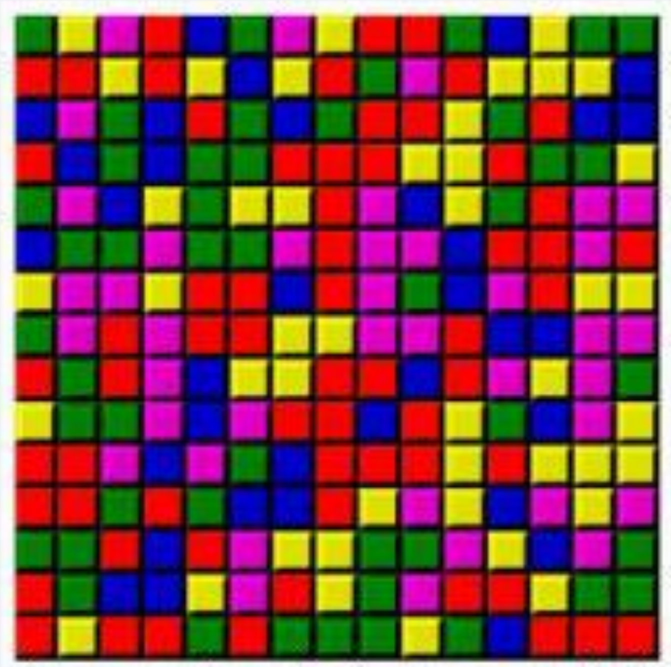


Figure. SameGame Puzzle.

- **State**: $n \times m$ grid,
  - **initial state**: initial placement of nm tiles.
  - **goal state**: empty grid.
- **Action**: legal removal of ***a group of tiles***:
  - a set of ≥ 2 tiles that are connected, and
  - of the same colour.
- **Transition model**: apply gravity and column shifting after removal of the group.

Problem 1

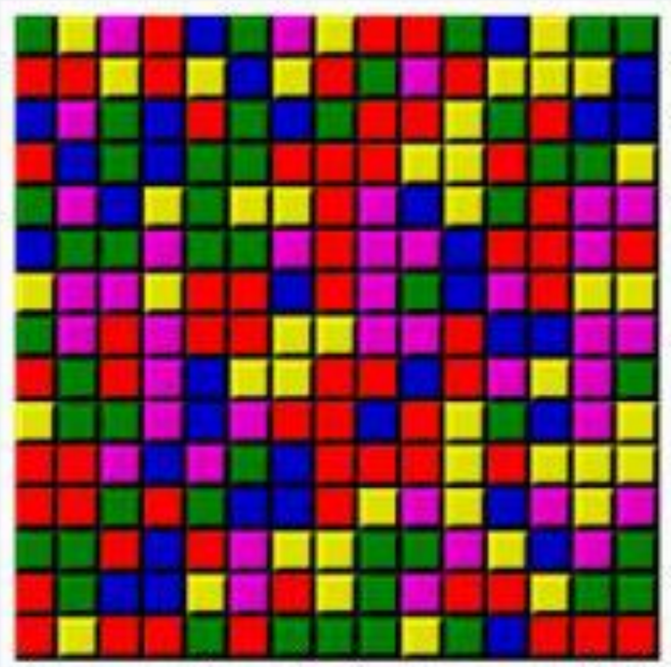
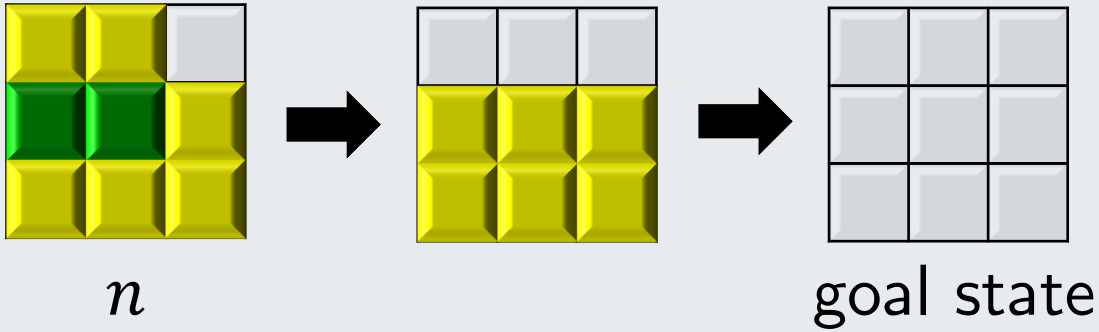


Figure. SameGame Puzzle.

- **Attempt**: $h(n)$ = number of groups.



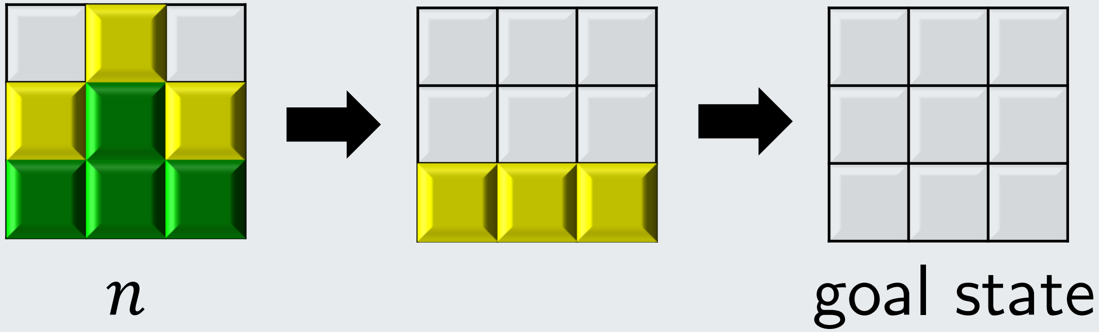
$h(n) = 3,\ h^*(n) = 2$

Actions can merge groups together.

$\Rightarrow$ not admissible!

Problem 1

- **Attempt**: $h(n)$ = number of singletons.



$h(n) = 3,\ h^*(n) = 2$

Again, actions can merge singletons together.

$\Rightarrow$ not admissible!

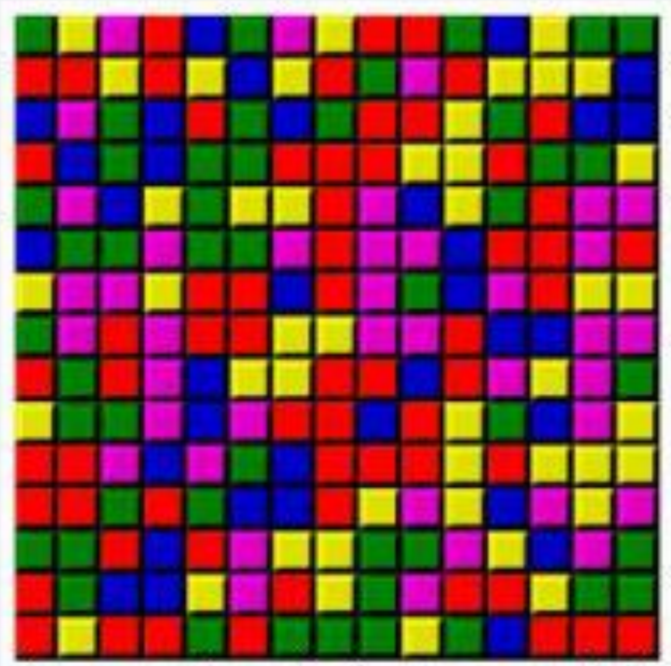


Figure. SameGame Puzzle.

Problem 1

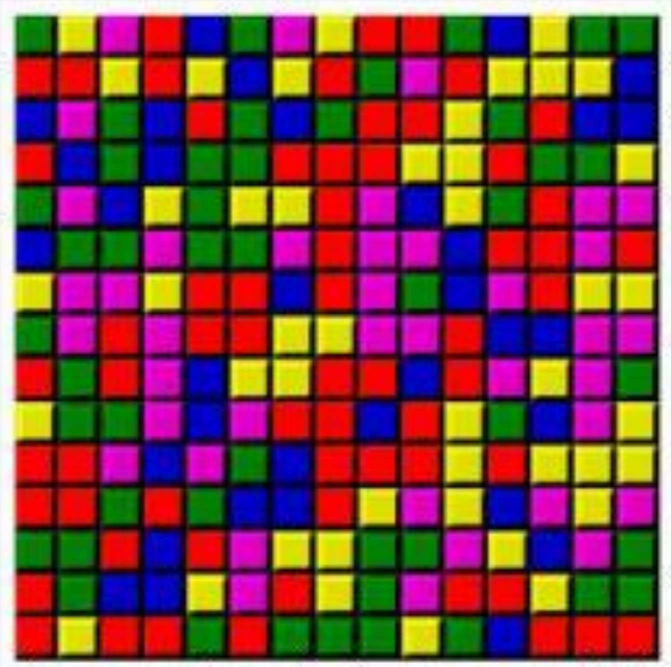


Figure. SameGame Puzzle.

- **State**: $n \times m$ grid,
  - **initial state**: initial placement of nm tiles.
  - **goal state**: empty grid.
- **Action**: legal removal of ***a group of tiles***:
  - a set of ≥ 2 tiles ~~that are connected,~~ and
  - of the same colour.
- **Transition model**: apply gravity and column shifting after removal of the group.

Optimal cost = number of remaining colours.

Problem 2

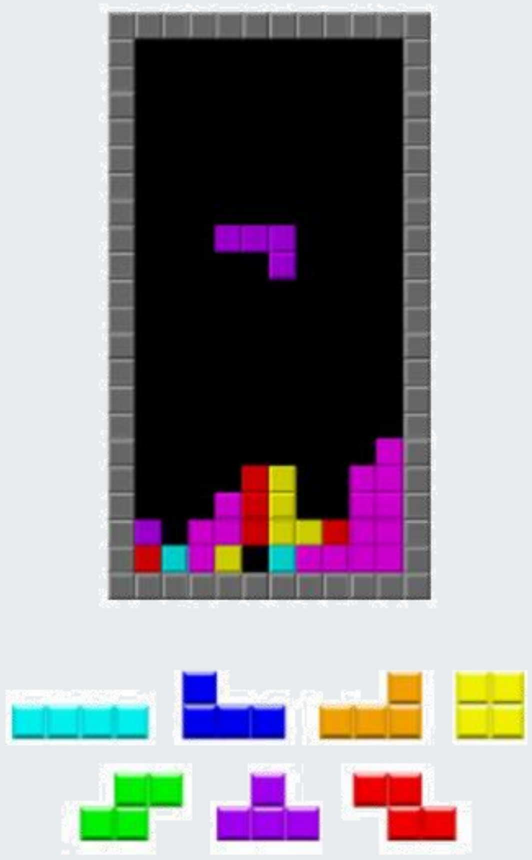


Figure. Tetris Puzzle.

- **State**: the Tetris field, and the next piece.
  - **initial state**: empty field.
  - **goal state**: completely filled field with no gaps after placing all N pieces.
- **Action**:
  choose the orientation and position of the next piece from where it descends.
  - **Action cost**: 1.
- **Transition model**: returns the field after the piece descended onto the field.

Problem 2.a

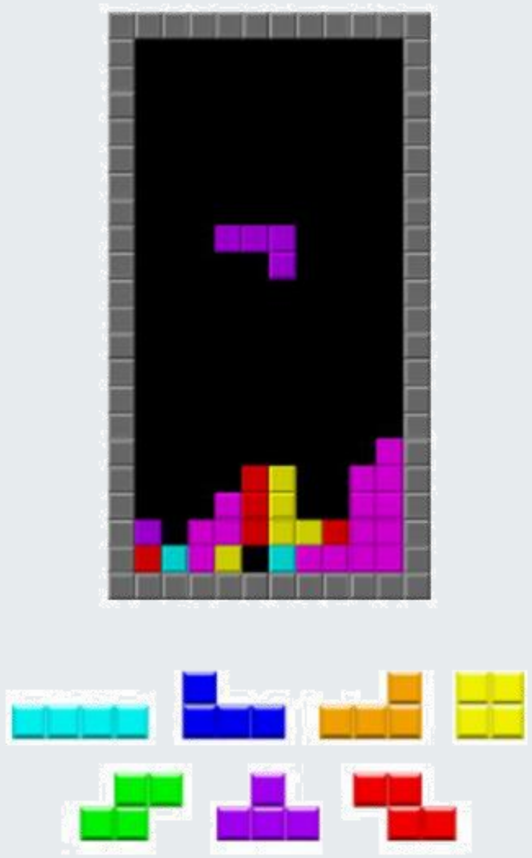


Figure. Tetris Puzzle.

- $h_1(n) =$ number of unfielded tetriminos.

Admissible! because we will need to place all the tetriminos to get to a goal state.

This is the *"sub-problem"* strategy:

- **State**: the Tetris field, and the next piece.
  - **initial state**: empty field.
  - **goal state**: ~~completely filled field with no gaps~~ after placing all N pieces.

Question: what is the problem with this heuristic?

$g(n) + h(n) = N$ for all states...

Problem 2.a

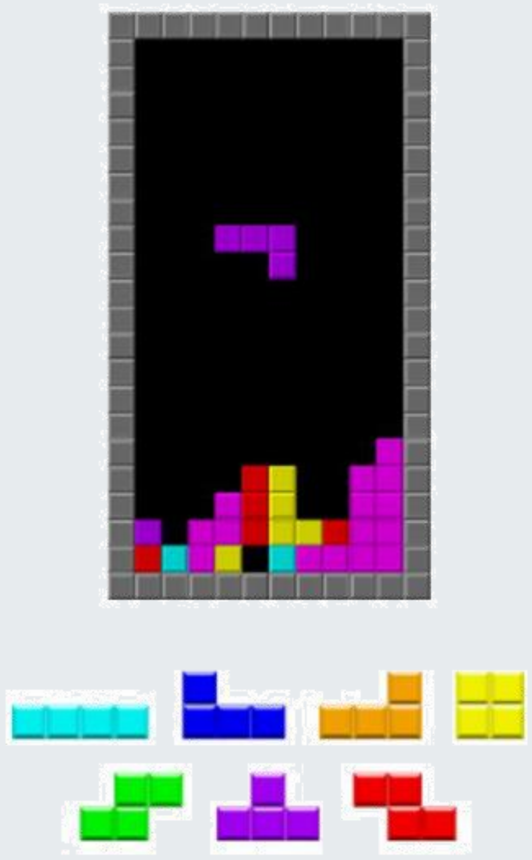


Figure. Tetris Puzzle.

- $h_2(n)$ = number of gaps.

Note. A gap is an empty cell on the board

Not admissible! because we can fill multiple gaps with one action.

Question: How can we make it admissible?

Use $h_2(n)/4$, since each piece fills 4 gaps.

Problem 2.a

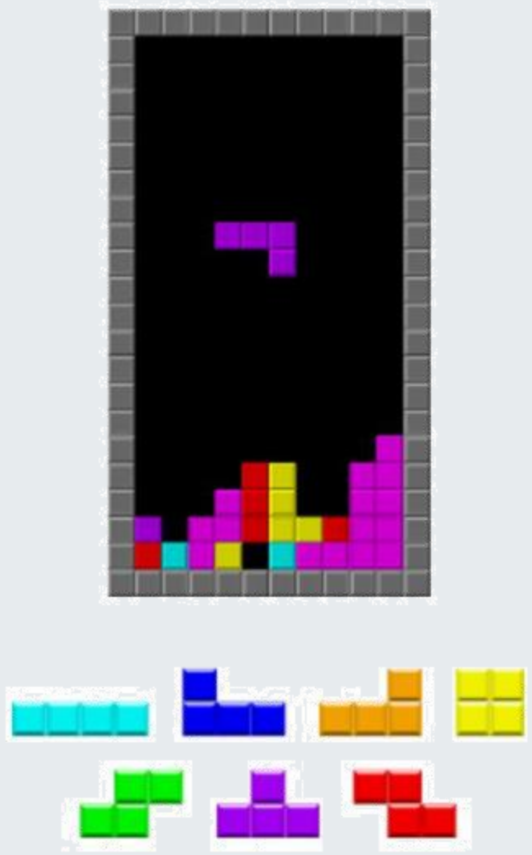
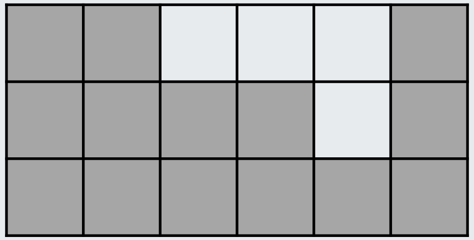


Figure. Tetris Puzzle.

- $h_3(n)$ = number of incomplete rows.

Note. A row is not complete if there's a gap in the row.

Not admissible! because we can make multiple rows complete with one action.



Question: How can we make it admissible?

Use $h_3(n)/4$, since each piece completes at most 4 rows.

Problem 2.a

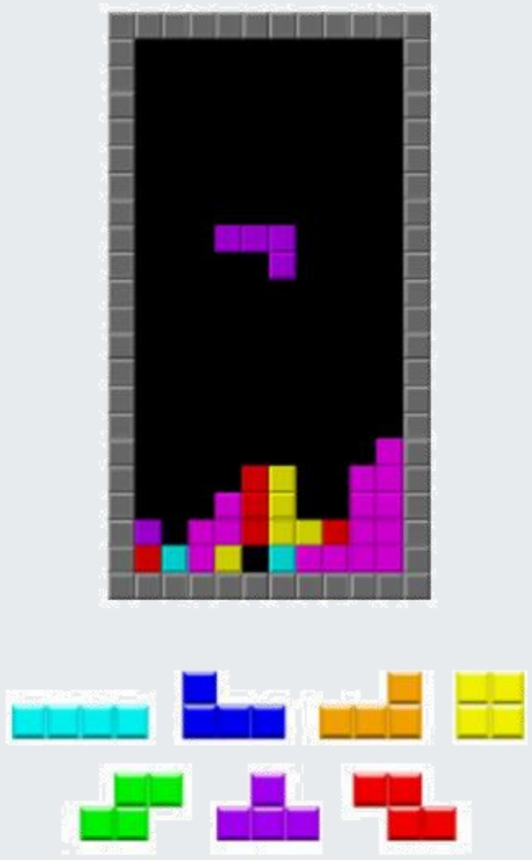


Figure. Tetris Puzzle.

- $h_4(n)$ = number of blocked gaps.

Note. A blocked gap is a gap that has an occupied cell above it.

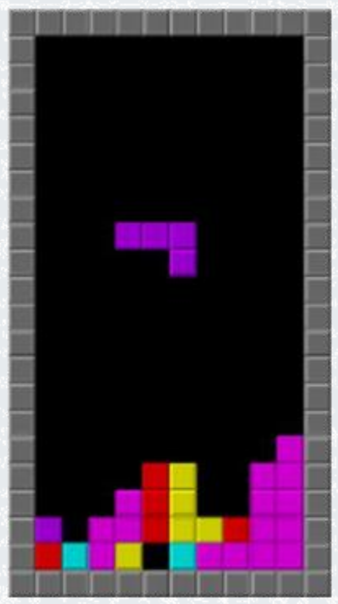
Admissible! If $h_4(n) > 0$, then $h^*(n) = \infty$.

This is the *"learn from experience"* strategy:

- **State**: the Tetris field, and the next piece.
  - **initial state**: empty field.
  - **goal state**: completely filled field with no gaps after placing all N pieces.

We will never reach goal if a gap is blocked.

Problem 2.b



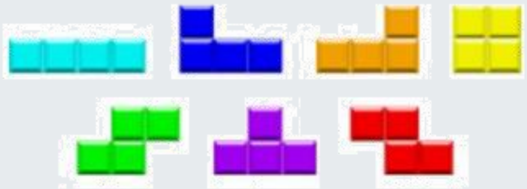


Figure. Tetris Puzzle.

- $h_1(n)$ = number of unfielded tetriminos. ✓
- $h_2(n)$ = number of gaps. ✗
- $h_3(n)$ = number of incomplete rows. ✗
- $h_4(n)$ = number of blocked gaps. ✓

- $\max\{h_1, h_2\}$. ✗
- $\min\{h_2, h_3\}$. ✗
- $\max\{h_3, h_4\}$. ✗
- $\min\{h_1, h_4\}$. ✓

Pause and Ponder

Suppose h_1 and h_2 are admissible heuristics.

Question 1: Is $\min\{h_1, h_2\}$ admissible?

Question 2: Is $\max\{h_1, h_2\}$ admissible?

Suppose h_3 and h_4 are consistent heuristics.

Question 3: Is $\min\{h_3, h_4\}$ consistent?

Question 4: Is $\max\{h_3, h_4\}$ consistent?

** Feel free to discuss in the telegram group!*

Problem 2.c

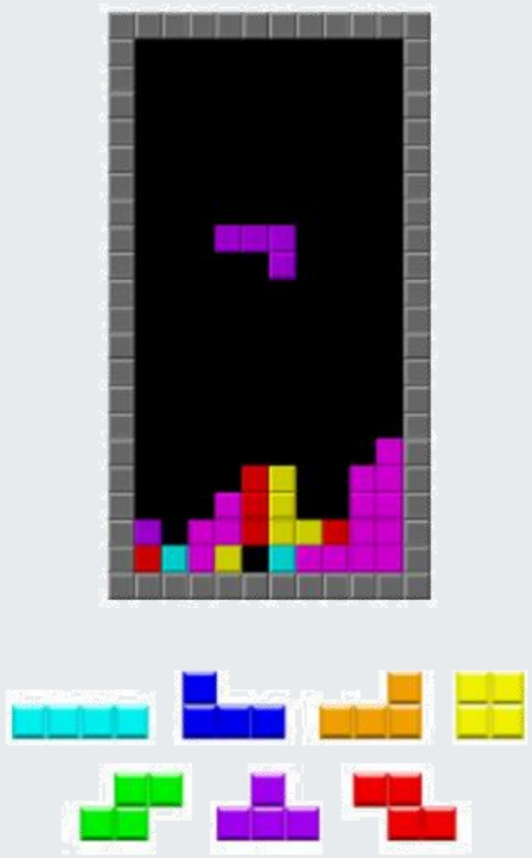


Figure. Tetris Puzzle.

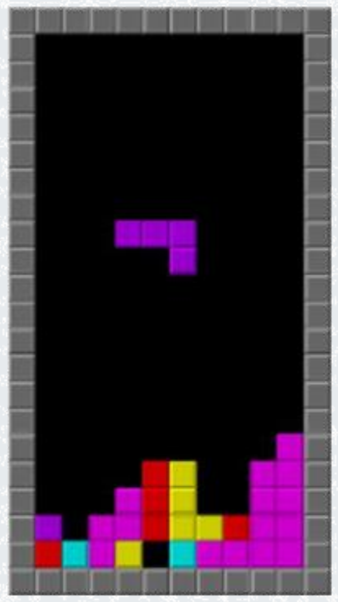
- $h_1(n)$ = number of unfielded tetriminos. ✓
- $h_2(n)$ = number of gaps. ✗
- $h_3(n)$ = number of incomplete rows. ✗
- $h_4(n)$ = number of blocked gaps. ✓

- h_1 dominates h_2.

No! because $h_2(n) = 4h_1(n)$ for all n.

** We are assuming dominance doesn't imply admissibility here.*

Problem 2.c



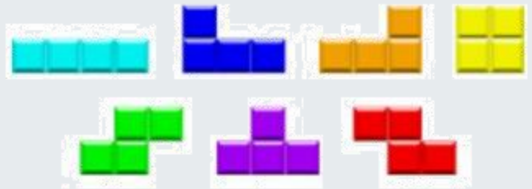


Figure. Tetris Puzzle.

- $h_1(n)$ = number of unfielded tetriminos. ✓
- $h_2(n)$ = number of gaps. ✗
- $h_3(n)$ = number of incomplete rows. ✗
- $h_4(n)$ = number of blocked gaps. ✓

- h_2 dominates h_4.

Yes! blocked gaps is just a subset of all gaps.

** We are assuming dominance doesn't imply admissibility here.*

Problem 2.c

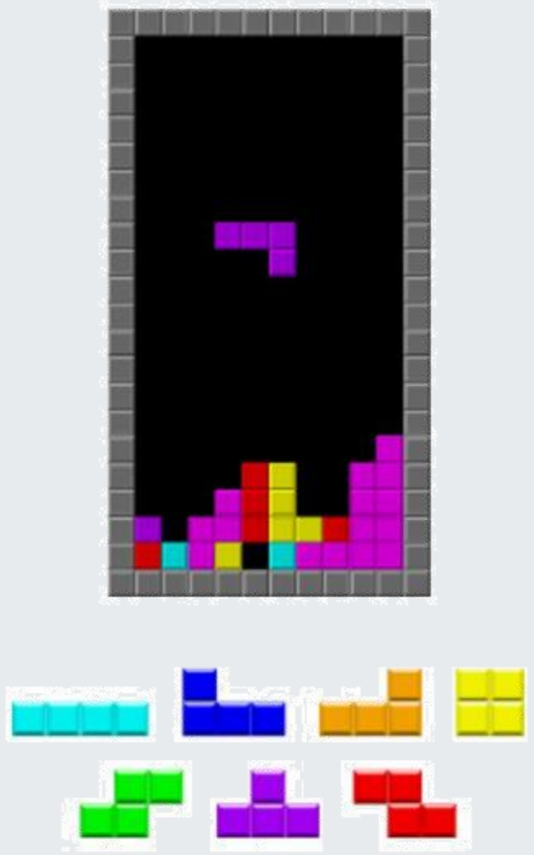
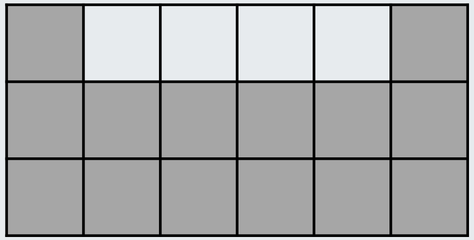


Figure. Tetris Puzzle.

- $h_1(n)$ = number of unfielded tetriminos. ✓
- $h_2(n)$ = number of gaps. ✗
- $h_3(n)$ = number of incomplete rows. ✗
- $h_4(n)$ = number of blocked gaps. ✓

- h_3 does not dominates h_2.



Yes! in this case $h_3 = 1$, $h_2 = 4$.

** We are assuming dominance doesn't imply admissibility here.*

Problem 2.c

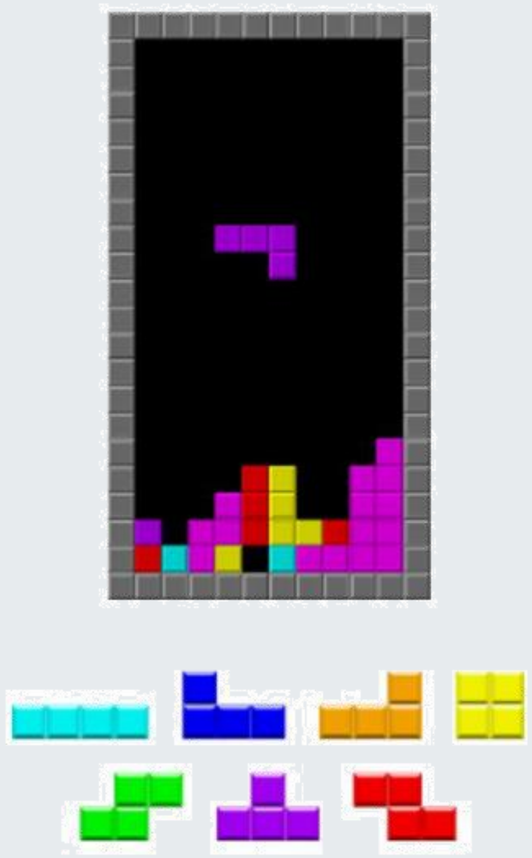
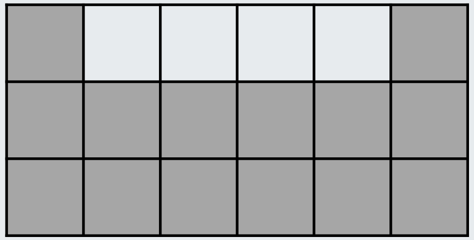


Figure. Tetris Puzzle.

- $h_1(n)$ = number of unfielded tetriminos. ✓
- $h_2(n)$ = number of gaps. ✗
- $h_3(n)$ = number of incomplete rows. ✗
- $h_4(n)$ = number of blocked gaps. ✓

- h_4 does not dominates $h_2/2$.



Yes! in this case $h_4 = 0$, $h_2/2 = 2$.

** We are assuming dominance doesn't imply admissibility here.*

Summary

Idea 1: Find the ***optimal cost*** of a relaxed (simplified) problem.

The closer to the original environment, the better.

Strategies:

- Relax the rules.
- Solve a sub-problem.
- Learn the heuristics*.

Idea 2: Combine several heuristics together.

If h_1, h_2 admissible/consistent, then $\max\{h_1, h_2\}$ is also admissible/consistent and dominates h_1 and h_2.

Question: what about $h_1 + h_2$?

** Learning agents, not in the scope of CS3243.*

Comparing Heuristics

Which heuristic function do we choose?

Problem 4

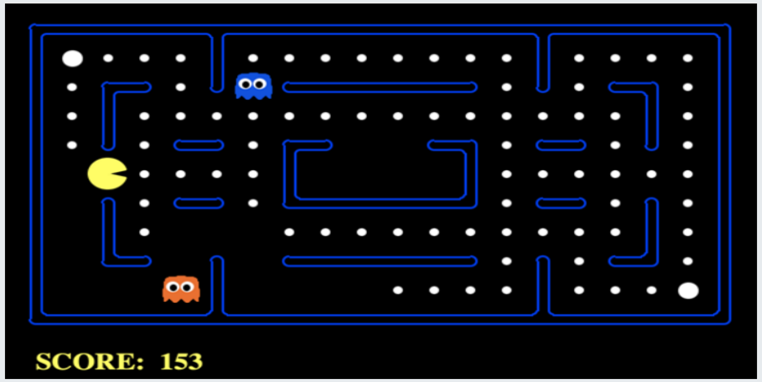


Figure. Pac-man.

- **State**: position of pac-man and positions remaining pallets.
  - **Initial state**: starting position of pac-man and grid entirely filled with pallets.
  - **goal state**: no pallets left on field.
- **Action**: move up/down/left/right.
  - **Action cost**: 1.
- **Transition model**: returns the field after the pac-man moved and consumed pallet.

Problem 4

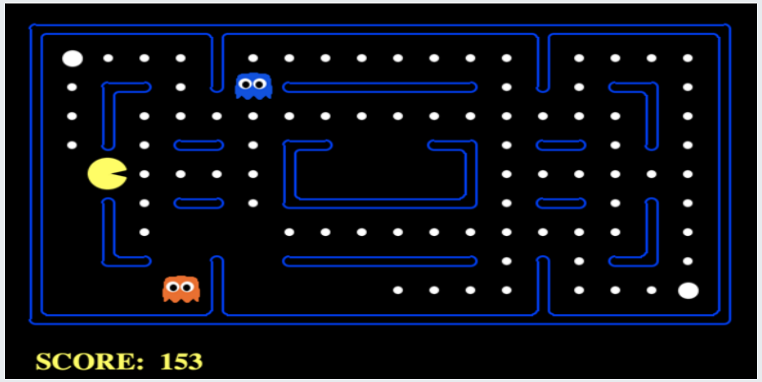


Figure. Pac-man.

- $h_1(n) =$ # remaining pallets,
- $h_2(n) =$ # remaining pallets + Manhattan distance to closest pallet,
- $h_3(n) =$ maximum Manhattan distance to a remaining pallet,
- $h_4(n) =$ average Euclidean distance to all pallets.

Claim: h_2 dominates h_1 *.

Since Manhattan distance to closest pallet ≥ 0.

** We are assuming dominance doesn't imply admissibility here.*

Problem 4

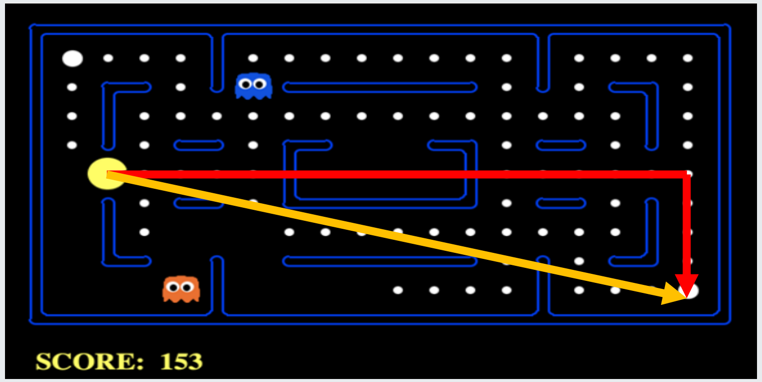


Figure. Pac-man.

- $h_1(n) = \#$ remaining pallets,
- $h_2(n) = \#$ remaining pallets + Manhattan distance to closest pallet,
- $h_3(n) =$ maximum Manhattan distance to a remaining pallet,
- $h_4(n) =$ average Euclidean distance to all pallets.

Claim: h_3 dominates h_4 *.

"Maximum" ≥ "average", "Manhattan distance" ≥ "Euclidean distance".

** We are assuming dominance doesn't imply admissibility here.*

End of File

Thank you very much for your attention!

References

- D. Ler, “Heuristics”, 2023. [Online].
- S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach," 3rd ed., Prentice Hall, 2010.