

HOMEWORK 01

Courses **CSC14003** : Intro to Artificial Intelligence

18CLC6, FIT - HCMUS.

17/07/2020

- **18127231** : Đoàn Đình Toàn (GitHub: [@t3bol90](#))
-

Problem 1. (2.0pts)

Give a PEAS description for each of the following activities

a. A tailor is sewing clothes on the sewing machine.

Assume that a tailor is a human agent, the PEAS description for this agent will be as follows:

- **Performance:** The Performance factor for a tailor will be the Correctness of sewing, Safety, Productivity, Time, etc.
- **Environment:** The Type of fabric, Style of clothes, Type of sewing machine, Place where this tailor work (at home, factory, ...), the Sequence of task (complete clothes or just a module), ...
- **Actuators:** The actuators is the part do the action of tailor: Hands, Legs, and other body part.
- **Sensors:** All 4 senses of tailor (not Taste), but Sight and Touch are the most importance senses for sewing. The Tailor need to see where the needle of sewing machine, using their hand and leg to control the machine (touch to know), feeling the type of fabric, change the form of clothes for some anchors, symbols,... Sometimes tailor can Smell or Hear the broken signal of machine, some power issue (if the machine is electrical sewing machine).

b. A customer is ordering food on GrabFood by using his/her smartphone.

Assume that our customer is a human agent, the PEAS description for this agent will be as follows:

- **Performance:** The Taste of food (good or bad), Prices, Feelings, Time, Experiences, etc.
- **Environment:** GrabFood application is an environment (it's fast, updated to newest version, outdated or not, supported or not,...), the Location of customer, Speed of the Internet, Type of his/her smartphone, etc.
- **Actuators:** Hands, Mind, maybe some part of body that can 'touch' to smartphone's screen.
- **Sensors:** Eyes, Ears (some restaurants have video as ads), Nose (in case our customer smell somethings and feel hungry :<), etc.

Problem 2. (2.0pts)

Examine the AI literature or the Internet to discover to what extent each of the following tasks can currently be solved by computers/robots:

a. Real time translation of spoken languages, e.g., from English to French, by Google Translate

Extention: Real time translation have been embed in many application and device (Skype Translator calling, Microsoft Translator, Google Translator - Google Assistant, WT2 Language Translator..). But it's performance sometimes does not good enough in some case and complex

languages - hard accents because of some NLP's problem in that languages haven't solved yet, eg: Google Translate can not translate correctly 'con cua có càng lớn và càng bé' from Vietnamese to English.

Rating: 3.5/5

References:

- [Amazon](#)
- [WT2](#)
- [Voice interfaces for real-time translation of common tourist conversation](#)

b. Fingerprint recognition in smart devices and laptops

Extention: Fingerprint recognition was one of the first techniques used for automatically identifying people and today is still one of the most popular and effective biometric techniques. TouchID of Apple (use for app/security authentication), Thinkpad's fingerprint are good example of how far fingerprint recognition have been. It's fast, high reliability but also have some potential security risk.

Rating: 4.5/5

References:

- [Fingerprint Recognition, Overview](#)
- [ngerprint Recognition](#)

c. The mastering of board games, including chess, Go and Shogi, by AlphaZero

Extention: Alpha-Zero system learned to play three challenging games (chess, shogi, and Go) at the highest levels of play seen. It was train on TPUs via process of trial and error called reinforcement learning and the important thing that many professional of board games include chess, Go, and Shogi said that AlphaGo is creative. In chess, AlphaZero won in all matches at 2016 TCEC, winrate 91.2% in shogi and 61% in Go.

Ratings: 5/5.

References:

- [AlphaZero: Shedding new light on chess, shogi, and Go](#)
- [Mastering board games](#)
- [DeepMind's AlphaZero beats state-of-the-art chess and shogi game engines](#)

d. Autonomous spaceport drone ship (ASDS) by SpaceX.

Extention: Autonomous spaceport drone ship (ASDS) by SpaceX is an ocean-going vessel derived from a deck barge, outfitted with station-keeping engines and a large landing platform and is controlled by an autonomous robot. It's a innovative thinking from SpaceX when build ASDS allow SpaceX to land boosters at sea on high-velocity missions that cannot carry enough fuel to allow for a return-to-launch-site landing. It will be a opening door for safety space travel and exploring the universe.

Ratings: 4/5. - It's a beginning, but it work well after many times trying fail/success and testing.

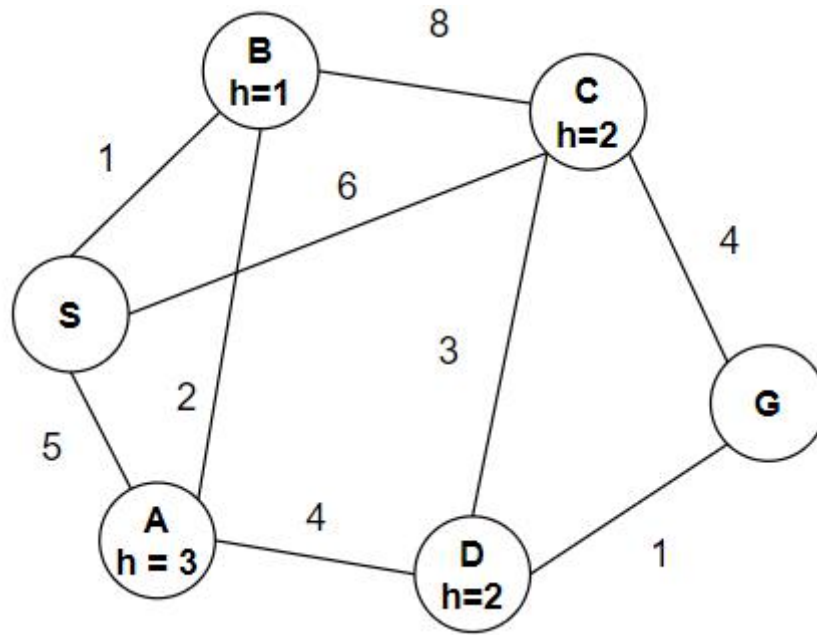
References:

- [Wikipedia](#)
- [Reddit](#) - I do not want to put it here but it's good references to another paper/references for more detail.

Write down your findings in 2-3 sentences each. The finding should rate the level of intelligence required on the task, on a scale of 1 (lowest) - 5 (highest), and then give explanation.

Problem 3. (3.0pts)

Consider the following graph. The start and goal states are S and G, respectively.



For each of the following graph search strategies, work out order in which states are expanded, as well as the path returned. In all cases, assume ties resolve in such a way that states with earlier alphabetical order are expanded first. Remember that in graph search, a state is expanded once.

Define:

- (v, w) : notation for node v behind node u one step in the optimal path with weight = w . (weight depend on the algorithm)
- $(v, w)^*$: expanding complete.
- x or 'null': not expanded.
- g : reach goal state.

a. Depth-first search.

Apply Depth-first search on the graph have this table bellow:

Step	S	A	B	C	D	G
1	(S,0)*	(S,1)	(S,1)	(S,1)	x	x
2		(S,1)*	(A,2)	(S,1)	(A,2)	x
3			(A,2)*	(B,3)	(A,2)	x
4				(B,3)*	(C,4)	(C,4)
5					(C,4)*	(D,5)
6						g

Expanding order: S, A, B, C, D, G.

Path: S -> A -> B -> C -> D -> G (cost = 19).

b. Breadth-first search.

Apply Breadth-first search on the graph have this table bellow:

Step	S	A	B	C	D	G
1	(S,0)*	(S,1)	(S,1)	(S,1)	x	x
2		(S,1)*	(S,1),(A,2)	(S,1)	(A,2)	x
3			(S,1)*	(S,1),(B,2)	(A,2)	x
4				(S,1)*	(A,2),(C,2)	(C,2)
5					(A,2)*	(C,2),(D,3)
6						g as (C,2)*

Expanding order: S, A, B, C, D, E.

Path: S -> C -> G (cost = 10).

c. Uniform cost search.

Apply Uniform cost search.

Step	S	A	B	C	D	G
1	(S,0)*	(S,5)	(S,1)	(S,6)	x	x
2		(B,3)	(S,1)*	(S,6)	x	x
3		(B,3)*		(S,6)	(A,7)	x
4				(S,6)*	(A,7)	(C,10)
5					(A,7)*	(D,8)
6						g

Expanding order: S, B, A, C, D, G.

Path: S -> B -> A -> D -> G, (cost = 8).

d. Greedy search with the heuristic h shown on the graph.

Apply Greedy search with h on graph.

Step	S	A	B	C	D	G
1	(S,0)*	(S,3)	(S,1)	(S,2)	x	x
2		(S,3)	(S,1)*	(S,2)	x	x
3		(S,3)		(S,2)*	(C,2)	(C,2)
4		(S,3)			(C,2)*	(C,2)
5		(S,3)				g as (C,2)

Expanding order: S, B, C, D, G.

Path: S -> C -> G, (cost = 10).

e. A* search with the same heuristic.

Apply A-star with $f = h + g$.

Step	S	A	B	C	D	G
1	(S,0)*	(S,8)	(S,1)	(S,8)	x	x
2		(B,6)	(S,2)*	(S,8)	x	x
3		(B,6)*		(S,8)	(A,9)	x
4				(S,8)*	(A,9)	(C,10)
5					(A,9)*	(D,8)
						g

Expanding order: S, B, A, C, D, G.

Path: S -> B -> A -> D -> G (cost = 8).

Problem 4. (3.0pts)

You are given the initial state (a) and the goal state (b) of an 8-puzzle as shown below:

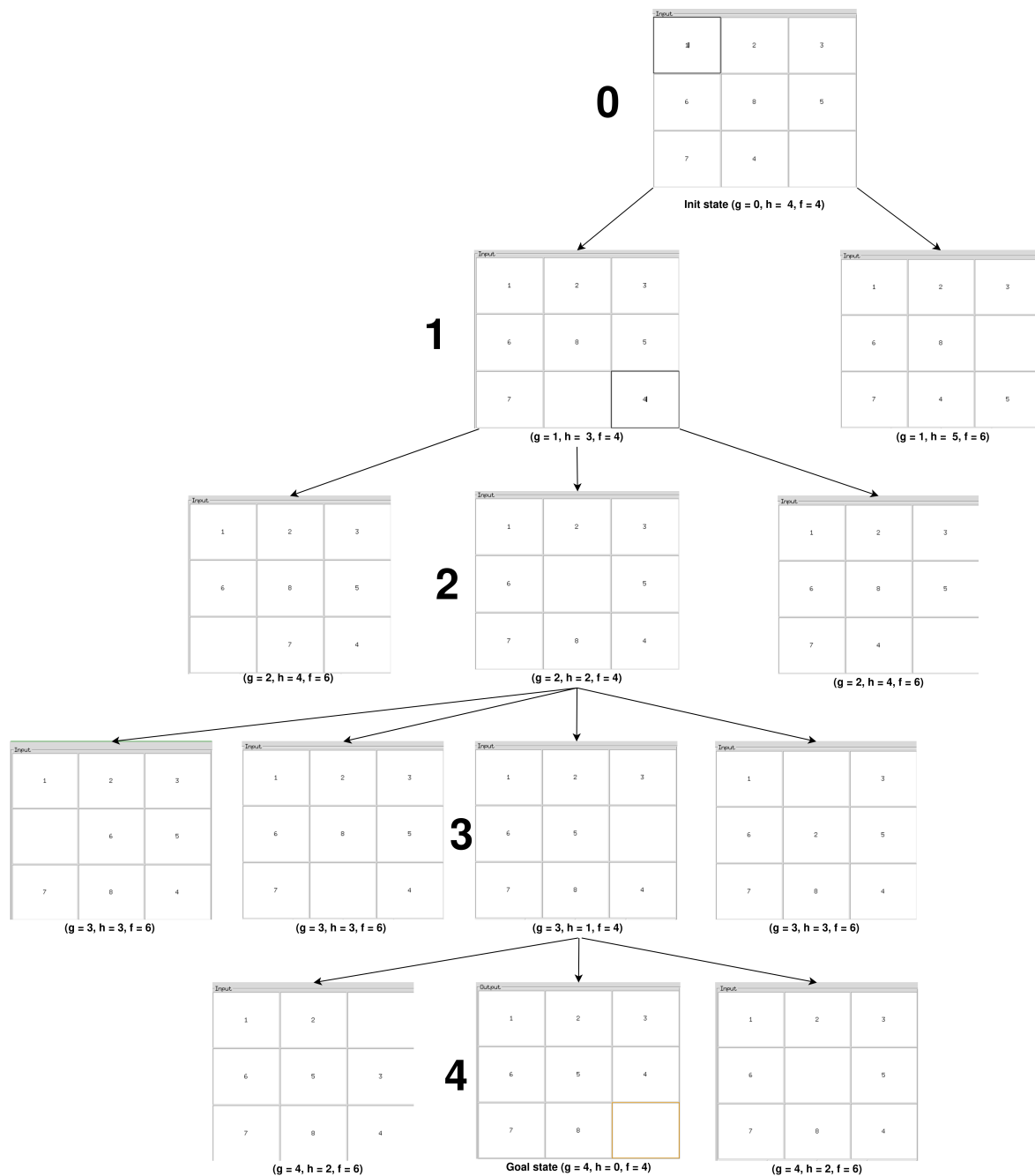
1	2	3
6	8	5
7	4	

(a)

1	2	3
6	5	4
7	8	

(b)

a. Apply A* using Manhattan distance heuristic function. Draw the search tree including possible expanded states during the algorithm procedure. Compute the triple (g, h, f) for each state. Mark the optimal strategy found.



Assume that we can go back to previous state when expanding. Sorry because small font size of tkinter.

For HD resolution: [drawio viewer](#), [png 200%](#).

b. For each of the following heuristics $h(n)$ (where n is the state) for the N-puzzle problem, identify its admissibility and justify your answer.

- $h(n) = 1$.

Answer: $h(n) = 1$ is not admissible because at goal state, $h(n)$ must be zero.

- $h(n)$ = the number of misplaced tiles at state n .

Answer: h is an admissible heuristic, since it is clear that every tile that is out of position must be moved at least once. $h(n) \leq h^*(n)$.

Proof: Let k be the misplaced tiles as state n . So $h(n) = k$ and we need to prove $h(n) \leq h^*(n)$.

In $n + 1$ state, there are 3 case can happen:

- No thin

- **$h(n)$ = (number of tiles out of row + number of tiles out of column).**

Answer: h is an admissible heuristic, since every tile that is out of column or out of row must be moved at least once and every tile that is both out of column and out of row must be moved at least twice. $h(n) \leq h^*(n)$.

===== END =====