

Homework #1

2022. 2학기. 인공지능

제출일: 2022/09/29

1. (Introduction) 인공지능의 개념과 관련하여 다음에 대해 설명하시오. 참고한 자료가 있으면 같이 제시하시오.

1) “Symbolic AI” 또는 “GOFAI” 는 무엇을 의미하는지 설명하시오.

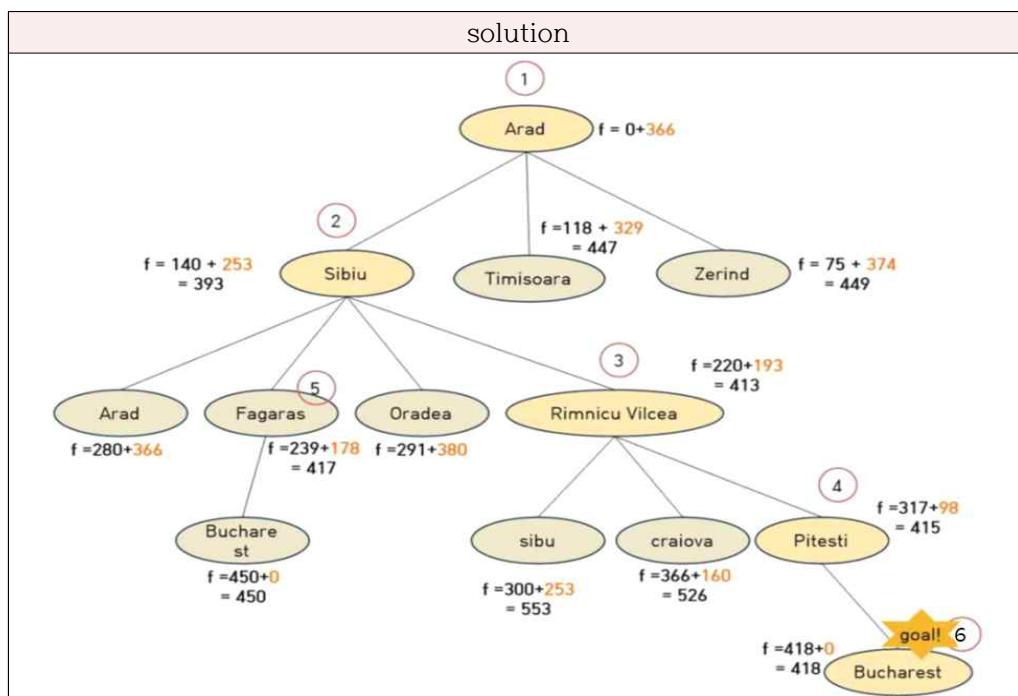
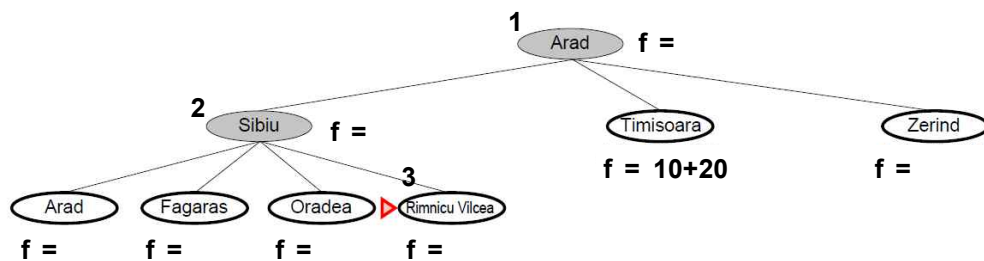
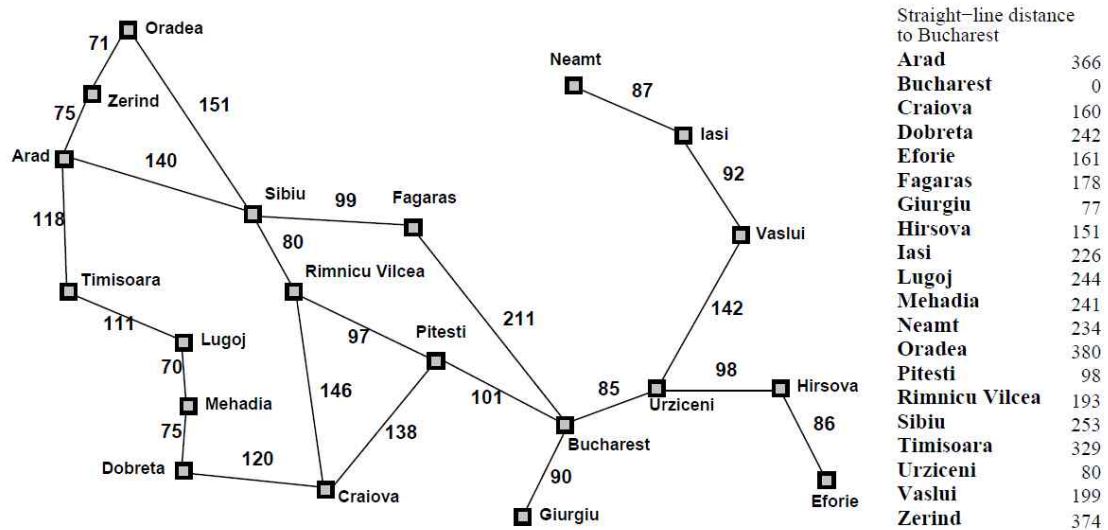
solution
<ul style="list-style-type: none">✓ Symbolic AI is the term for the collection of all methods in artificial intelligence research that are based on high-level symbolic representations of problems, logic and search.✓ Symbolic AI developed applications such as knowledge-based systems (expert systems), automated theorem provers, ontologies, semantic web, automated planning, etc.✓ GOFAI is an acronym for "Good Old-Fashioned Artificial Intelligence".✓ GOFAI refers a kind of symbolic AI, namely rule-based or logical agents.

2) “Symbolic AI” 와 다른 AI 구현 방식은 어떤 것이 있는지 말해보시오.

solution
<ul style="list-style-type: none">✓ Connectionist AI approaches include techniques and algorithms using artificial neural networks.✓ Connectionist principle is that mental phenomena can be described by interconnected networks of simple units, as in human brain.✓ Connectionism presents a cognitive theory based on simultaneously occurring, distributed signal activity via connections that can be represented numerically, where learning occurs by modifying connection strengths based on experience.

2. (Search) 다음의 지도는 루마니아 어느 지역의 도시간 도로상 거리를 나타낸다. 우측에 명시된 숫자는 각 도시와 수도인 Bucharest와의 직선 거리이다.

1) Arad에서 Bucharest 까지의 최단 경로를 A* 알고리즘으로 찾는 과정을 아래 트리과 같이 그려보시오. 각 노드에 방문하는 순서와 해당 state의 평가값 f를 명시하시오.



- 2) 찾고자 하는 것이 최단 경로가 아닌 최소 시간 경로라고 하자. 각 도시간 이동 시간이 주어졌을 때, A* 알고리즘을 적용하기 위한 heuristic은 어떻게 정의할 수 있겠는가?

solution
탐색의 결과로 얻은 path가 최소시간경로임을 보장하려면, 휴리스틱 함수 $h(n)$ 이 n 에서 goal까지 추정 이동시간, $h^*(n)$ 이 n 에서 goal까지 실제 이동시간이라고 할 때 다음을 만족해야 한다.
$h(n) \leq h^*(n)$
따라서 $h(n)$ = “ n 과 goal 사이에 직선 고속도로가 있다고 가정할 때의 이동 시간” 으로 정의할 수 있다.

3. (Search) 아래 그림은 어떤 블록쌓기 문제의 시작상태와 목표상태를 나타낸다. 블록이 쌓여져 있는 pile들의 구성에 따라 서로 다른 상태가 되고, 쌓여진 pile 간의 상대적인 위치는 중요하지 않다고 하자. 즉, (2, 5, 3)과 (1, 4)가 있는 것과 (1, 4)와 (2, 5, 3)이 있는 것은 같은 상태이다. 시작상태에서 최소한의 블록 이동으로 목표상태에 도달하는 방법을 찾고자 한다.



- 1) 이 문제를 state space로 표현하기 위해 state, action, start, goal을 정의하시오.

solution
<ul style="list-style-type: none"> state : 블록이 쌓여져 있는 상태 action : (위에 아무것도 없는) 하나의 블록을 다른 위치로 이동시키는 것. start : (2,5,3), (1,4) goal : (1,2,3,4,5)

- 2) Heuristic $h(n)$ 을 다음과 같이 정의한다고 하자.

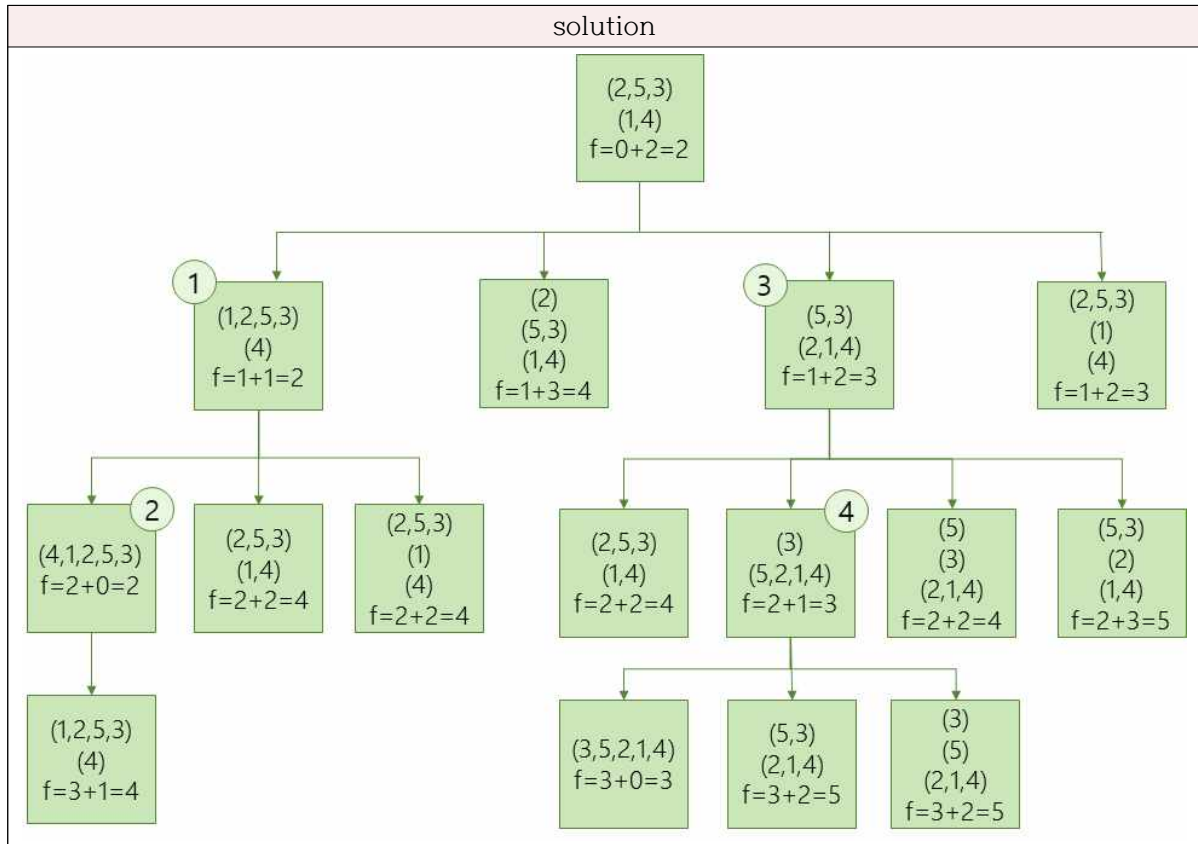
$h(n)$ = 가장 높게 쌓여진 pile 에 있지 않은 블록의 수

(예를 들어 $h(\text{시작상태}) = 2$, $h(\text{목표상태}) = 0$)

이러한 $h(n)$ 은 admissible 한가? 설명하시오.

solution
$h(n)$ = ‘가장 높게 쌓여진 pile 에 있지 않은 블록의 수’ 가 $h^*(n)$ = ‘목표 상태가 되기 위해 블록을 옮겨야 하는 횟수’보다 항상 같거나 작기 때문에 $h(n)$ 은 admissible 하다.

- 3) 위의 $h(n)$ 을 이용하여 A* search를 진행한다고 할 때, 처음 4개의 state를 expand하는 과정까지만 search tree를 그려보시오. 각 state에 expand된 순서와 $f(n)$ 을 명시하시오. 하나의 블록을 다른 위치로 옮기는 action의 cost는 1이다.



4. (Logic) 명제논리와 관련하여 다음에 답하시오.

- 1) 논리적 추론 과정이 “sound” 하다는 것은 무엇을 말하는지 설명하시오.

solution
논리적 추론 과정이 S로부터 X를 만들어낸다고 할 때, S가 참이면 X가 항상 참인 것을 말한다.

- 2) 아래와 같은 추론이 sound 한 것인지 아닌지 truth table을 통해 보이시오.

$$P \vee Q$$

$$\sim Q \vee R$$

⊢

$$P \vee R$$

solution						
P	Q	R	$\sim Q$	$P \vee Q$	$\sim Q \vee R$	$P \vee R$
T	T	T	F	T	T	T
T	T	F	F	T	F	T
T	F	T	T	T	T	T
T	F	F	T	T	T	T
F	T	T	F	T	T	T
F	T	F	F	T	F	F
F	F	T	T	F	T	T
F	F	F	T	F	T	F

Sound 하다.

▪ $(P \vee Q)$ 와 $(\sim Q \vee R)$ 이 모두 T 일 때, $(P \vee R)$ 은 항상 T 이다.

3) 다음과 같은 명제들 P, Q, R, S, T에 대해 아래 문장들 s1 ~ s4 가 모두 참이라고 하자. 이 문장들을 logical sentence들로 표현하고, 추론규칙들을 순차적으로 적용하여 명제 T 가 참임을 추론하는 과정을 상세히 써보시오.

P: It's sunny this afternoon.

Q: It's colder than yesterday.

R: We will go camping.

S: We will visit the zoo.

T: We will be home by sunset.

s1: "It's not sunny and it's colder than yesterday"

s2: "We will go camping only if it's sunny"

("If we will go camping then it's sunny")

s3: "If we don't go camping then we will visit the zoo"

s4: "If we visit the zoo, then we will be home by sunset"

solution
s1: "It's not sunny and it's colder than yesterday"
① $\sim P \wedge Q$
s2: "We will go camping only if it's sunny" ("If we will go camping then it's sunny")
② $R \Rightarrow P$
s3: "If we don't go camping then we will visit the zoo"
③ $\sim R \Rightarrow S$
s4: "If we visit the zoo, then we will be home by sunset"
④ $S \Rightarrow T$
Apply And Elimination to ① \rightarrow ⑤ $\sim P$, ⑥ Q
Apply Modus Tolens to ②, ⑤ \rightarrow ⑦ $\sim R$
Apply Modus Ponens to ③, ⑦ \rightarrow ⑧ S
Apply Modus Ponens to ④, ⑧ \rightarrow ⑨ T
\therefore "We will be home by sunset"

5. (Programming) 8 puzzle을 위한 A* search 프로그램을 작성하고 다음 탐색 과정을 수행하시오.
제공되는 파이선 프로그램 코드 8-puzzle-sample-code.ipynb 를 참고하시오.

- 1) 코드의 빈 부분을 을 작성하여 아래와 같은 start, goal 에 대한 optimal path를
구하시오. ('0' 이 빈칸을 의미)

Start: 123804765

Goal: 781602543

1 2 3
8 0 4
7 6 5



7 8 1
6 0 2
5 4 3

solution	
<pre>def ASTAR(startNode): ... while not(OPEN.empty()): # Choose the best node from OPEN currentNode = OPEN.get()[1] print(".", end='') # If it is in CLOSED, ignore if currentNode.state not in CLOSED: # Test if it is a goal if goaltest(currentNode.state): return currentNode # Expand the node else: CLOSED.append(currentNode.state) nodeList = expand(currentNode) for node in nodeList: if node.state in CLOSED: continue OPEN.put((node.f, node)) return None</pre>	
<pre># hamming distance def heuristic(state): h = 0 for i in range(puzzleSize): if (state[i] == 0): continue if (state[i] != goalState[i]): h = h + 1 return h</pre>	<pre># manhattan distance def heuristic(state): h = 0 for s in state: if (s == 0): continue r = abs(state.index(s)/3 - goalState.index(s)/3) c = abs(state.index(s)%3 - goalState.index(s)%3) h += c + r return h</pre>

- 2) 다음의 heuristic 함수를 이용했을 때 탐색이 어떻게 달라지는지 비교하고 그 이유를 설명하시오.

$h1$ = number of tiles out of place

$h2$ = sum of distances out of place

solution
h1의 결과: Start-U-L-D-D-R-R-U-U-L-L-D-D-R-R-U-L h2의 결과: Start-R-U-L-L-D-D-R-R-U-U-L-L-D-D-R-U
두가지 경우 모두 admissible heuristic으로서 최단 경로를 찾는다. 그러나 h1을 이용한 경우보다 h2를 이용한 경우에 훨씬 적은 수의 state를 방문하고 탐색이 완료되는 것을 확인할 수 있으며, 이는 항상 $h1 \leq h2 \leq h^*$ 이기 때문이다.