AI Planning for Autonomy

Problem Set I: Blind Search

1. Choose **one** of the problems listed below and describe a simple example along with its corresponding *State Model*.

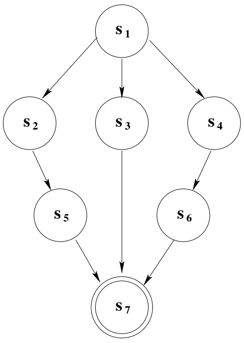
The problems are:

1. 8-Puzzle.
2. Travelling Salesman Problem.

Definition should be brief, clear, and compact [1]

- State space S
- Initial state $s_0 \in S$
- Set of *goal* states $S_G \subseteq S$
- Applicable actions function $A(s)$ for each state $s \in S$
- Transition function $f(s, a)$ for $s \in S$ and $a \in A(s)$
- Cost of each action $c(a, s)$ for $s \in S$ and $a \in A(s)$

2. Consider the following state space S, where $s_0 = s_1$ and $S_G = \{s_7\}$



where actions changing a state s into another state s' are given by the edges. The cost to transition from state s to s' is given by the following table:

s	s'	$c(s, s')$	s	s'	$c(s, s')$
s_1	s_2	2	s_3	s_7	10
s_1	s_3	2	s_4	s_6	1
s_1	s_4	1	s_5	s_7	3
s_2	s_5	2	s_6	s_7	4

[1] *Compact* means using mathematical notation to define sets, i.e. $S = \{x | x \in V\}$ to define that there are as many states as elements in the set V, and pseudo-code, i.e. to define the transition function.

Describe the execution of Breadth First Search ($BrFS$), Depth First Search (DFS) and Iterative Deepening (ID) in this problem by filling in a table like the one below. Show the order in which nodes are expanded. Each node must be *named*, e.g. $n_3 = \langle s_3, g(n), n_{parent}\rangle$. The node should contain all the relevant information for the search: current state s_i, the accumulated cost of the path from the initial state s_0 to s_i, and a pointer to the parent node.

	Breadth First Search
Ordered Sequence of States expanded Example: $\langle n_1 =< s_1, 0, - >, n_2 =< s_2, 2, n_1 >, \ldots\rangle$	

	Depth First Search -
Ordered Sequence of States expanded Example: $\langle n_1 =< s_1, 0, - >, n_2 =< s_2, 2, n_1 >, \ldots\rangle$	

	Iterative Deepening -
Ordered Sequence of States expanded Example: $\langle n_1 =< s_1, 0, - >, n_2 =< s_2, 2, n_1 >, \ldots\rangle$	

- Which is the solution found by each algorithm?
- Which is the optimal solution?
- Explain under which conditions the algorithms guarantee optimality?
- Adapt any of the previous algorithms to account for $g(n)$. Explain properties: optimality,complete,sound.