# CS 561 - ASSIGNMENT 1 SOLUTIONS

**Question 1**

**1. Chess player**

### Type - 1 (player type)
(a) PAGE description
      Percepts: the board, types of chess pieces and their movements
      Actions: move a piece on the board
      Goals: win the game
      Environment: Chess board

(b) Properties of the environment

| Accessible | Deterministic | Episodic | Static | Discrete |
|---|---|---|---|---|
| Yes | Yes | No | semi/yes | yes |

(c) Type of the problem
      Single - state problem

### Type – 2 (game monitor type – another possibility)

(a) PAGE description
      Percepts: reading from clock; the move made by each player
      Actions: giving a timeout for a player to move; preventing an illegal move; resolving disputes
            between the players if any arises
      Goals: Monitor the game to ensure legality / order
      Environment: Chess board

(b) Properties of the environment

| Accessible | Deterministic | Episodic | Static | Discrete |
|---|---|---|---|---|
| Yes | Yes | Yes | semi/yes | yes |

(c) Type of the problem
      Single - state problem

## 2. Robot navigation avoiding obstacles

(a) PAGE description
      Percepts: pictures from the camera to realize obstacles; depth/range information from sonar; etc.
      Actions: move, stop and change the direction of movement
      Goals: navigating without hitting obstacles (maybe from some source to some destination)
      Environment: maybe some predefined area

(b) Properties of the environment

| Accessible | Deterministic | Episodic | Static | Discrete |
|---|---|---|---|---|
| No | No | No | No | No |

(c) Type of the problem
      Contingency problem

## 3. Spelling and grammar assistant.

### Type 1 (An offline checker – It processes an entire document in a known language as a batch)

(a) PAGE description
- Percepts: characters, words, sentences
- Actions: replace, ignore, add to dictionary, underline, revise fragment …
- Goals: everything is correct in the text
- Environment: PC / Word processor / Editor …

(b) Properties of the environment

| Accessible | Deterministic | Episodic | Static | Discrete |
|---|---|---|---|---|
| Yes | Yes | Yes | Yes | Yes |

(c) Type of the problem
- Single-state problem

### Type 2 (An online checker – processes new keystrokes or words as they arrive)

(a) PAGE description
- Percepts: characters, words, sentences
- Actions: replace, ignore, add to dictionary, underline, revise fragment …
- Goals: everything is correct in the text
- Environment: PC / Word processor / Editor …

(b) Properties of the environment

| Accessible | Deterministic | Episodic | Static | Discrete |
|---|---|---|---|---|
| Yes/partially | Yes | Yes | Yes | Yes |

(c) Type of the problem
- Single-state problem/Multi-state problem

## 4. Mathematician's theorem-proving assistant.

(a) PAGE description
- Percepts: Given axioms and rules to be applied at each step, and the results that the assistant may find in the middle stages and the theorems.
- Actions: apply rule/axiom, generating output (lists of theorems) and manipulating symbols to create new results from old.
- Goals: Proof the theorem (or showing that there's no proof)
- Environment: PC / Prolog / Expert Systems / Any other rule based systems or theorem provers

(b) Properties of the environment

| Accessible | Deterministic | Episodic | Static | Discrete |
|---|---|---|---|---|
| Yes | Yes | Yes/No | Yes | Yes |

(c) Type of the problem
- Single-state problem

**5. A robot assembling a machine from parts**

(a) PAGE description

      Percepts: different parts of the machine, the machine (another more detailed view would be pixels of
            varying intensity)
      Actions: pick the part, put the part in the appropriate place (to assemble the whole machine)
      Goals: assembling the machine correctly
      Environment: assembly line

(b) Properties of the environment

| Accessible | Deterministic | Episodic | Static | Discrete |
|---|---|---|---|---|
| No | No/Yes | Yes/No | No/Yes | No/Yes |

(c) Type of the problem

      Contingency problem / multi-state problem.

---

**NOTE – Answers exactly same as these are awarded full points. 50 points with 5 questions, each
having 10 expected answers accounting for 10 points.**

**Please also note**
**(a)**
**Percepts = "sensory" input – and not an input device**
**Actions = what "effectors" can do – not the intelligence involved**
**And correspondingly proper goals and environment**

**(b)**
**Nothing special here – you just need to apply the text/class definitions with your assumptions**

**(c)**
**Again the text/class definitions need to be applied to the problem.**

---

**Question 2**

**1. Traveling salesman problem (TSP):**

<div align="center"><b>Type – 1 (standard type)</b></div>

      (a) Initial state: A list of cities Cost of getting from one city to another and the starting point.
      (b) Goal test: The salesman travels all the cities exactly once and returns to the starting city so that
            the total cost of covering all cities is minimum.
      (c) Successor function: From the current city, go to another city
      (d) Cost: Cost of each move is equal to the distance between source and destination.

<div align="center"><b>Type – 1 (a board description type)</b></div>

      (a) Initial state: A list of cities Cost of getting from one city to another and the starting point. A chart
            with all cities listed in the From axis and the To axis is provided. Only one half of the
            board will be used . The solution is formed by linking a from-to coordinate in the
            board. (Think of From and To axis like X/Y axis like a distance chart)
      (b) Goal test: A set of points on one half of the distance chart linking all the cities subject to the
            constraints of the problem
      (c) Successor function: Add a link between a "From" and a "To" cities
      (d) Cost: Cost of each move is equal to the distance between source and destination of move.

**2. Mine sweeper:**
    (a) Initial state: The minefield is all covered.
    (b) Goal test: All the mines are located correctly
    (c) Successor function: Uncover square, flag square
    (d) Cost: Some function of correct locations, incorrect locations and probably the time spent.


**3. Phone book**
    (a) Initial state: The phone book is open on page n (and the target lies on some unknown page).
    (b) Goal test: The agent finds the name of the person you're looking for.
    (c) Successor function:  either turn x pages
                            or turn 1 page (special case of above with x =1 )
                            or scan previous/next x entries (one the same page) at a time
                            or scan next/previous entry in the same page

    (d) Cost: Some function of number of scan operations (with varying cost for each type of scan) and
            also the time taken to reach the entry needed.

---

**NOTE –**
**The idea isn't to solve the problem – you are just being tested at being able to formulate a problem in a certain manner given the situation.**

**The successor/cost function is just an operator which when applied to a state generates all possible states you can go to from that state– so we are looking fo a simple operator (action) that can be applied to a state – answers that give strategies or the resulting states are incorrect.**

**Each part will carry 8 points (4 parts per question with 2 points for each) – total = 24 points (as against 25 stated in the assignment).**

---

**Question 3**

**1.** BFS works better, because BFS can always find the optimal solution as against DFS (although both have the same time complexity). In my specific implementation, BFS ran [faster/slower] and … [additional comments on how your code performed].

**2.** 100 – There are 100 squares and one game piece to place on these squares.  Note: we were not asking you the size of the search tree.

**Note –  Bonus +10pt**
There are 10x10=100 possible starting states, and 100 possible goal states (assuming start=goal is an allowed option); so there are only 10,000 start/goal combinations. So you can actually try all possible combinations to justify your code assumptions if any.

---

**NOTE –**
**The program by itself will carry 70 points and the two theory questions will carry 3 points each. Thus the total here will be 76 points (as against the 75 stated in the question).**

**The program can either be right or wrong - you can't claim that your program was more or less right if it gives some right and some wrong answers. So the marking for the program will be very non-linear (meaning that a correct program will be given full points but a "semi-wrong"/"fully" wrong program will loose a lot/all of points. We have decided to give more weightage to the BFS part to benefit the students.**

**Additionaly the following can be noted for your reference –**

**Total of 15 test cases listed below :-**

**For the BFS part –**          **all 7 correct => 40 marks**
                                    **>= 5 correct => 25 marks**
                                    **>= 3 correct => 20 marks**
                                    **0 correct   => 0 marks**

**For the DFS part –**          **all 8 correct => 30 marks**
                                    **>= 5 correct => 15 marks**
                                    **>= 3 correct => 10 marks**
                                    **0 correct   => 0 marks**

**Intermediate cases are given marks in the appropriate ranges**

Test cases –
         The format that was specified in the discussion forum – this is the ideal & correct answer
problem.txt
15
1 2 3 4 5
1 2 3 6 7
1 0 0 8 9
1 0 9 8 7
1 9 8 6 1
1 3 3 3 3
1 0 0 7 5
2 0 0 9 9
2 1 1 9 9
2 3 4 1 2
2 2 3 1 1
2 4 9 0 0
2 0 0 0 0
2 3 3 6 4
2 9 7 3 3

solution.txt
2
2
-1
4
3
0
3
5
4
2
-1
-1
0
1
4

Some people are still not members of the discussion forum – although the grader has tried to check, for as many cases as is possible, whether the students program works with the following input format

| Line Number | Contents: |
|---|---|
| 1. | 15 |
| 2. | 1 2 3 4 5 |
| 3. | 1 2 3 6 7 |
| 4. | 1 0 0 8 9 |
| 5. | 1 0 9 8 7 |
| 6. | 1 9 8 6 1 |
| 7. | 1 3 3 3 3 |
| 8. | 1 0 0 7 5 |
| 9. | 2 0 0 9 9 |
| 10. | 2 1 1 9 9 |
| 11. | 2 3 4 1 2 |
| 12. | 2 2 3 1 1 |
| 13. | 2 4 9 0 0 |
| 14. | 2 0 0 0 0 |
| 15. | 2 3 3 6 4 |
| 16. | 2 9 7 3 3 |

It may be possible that someone has got a zero for the programming part and may have actually used the second form – if this is the case please demo it to the grader (with the program you have already submitted) and your marks will be updated accordingly. **This format will be accepted only for this assignment. Please use the first format for all future assignments.**

In general, if you haven't got what you expected in the programming part, it is recommended that you try these cases with your programs in your aludra accounts. If it works there and there is still a discrepancy (very remote possibility) – please meet the grader – he will update your marks accordingly - after he reviews your work.

**Also please follow the following steps if you have any questions regarding this assignment**

**(1) If your answers match the scheme and you haven't got marks (or) if you have doubts regarding the coding part – please email the grader directly.**

**(2) If you have questions regarding the theory part – please speak with your instructor and ask him to make the changes on your hard-copy – after this step – please bring the hardcopy to the grader to get your marks updated.**

**(3) All those who are not yet members of the discussion forum – please note that you are loosing out on lots of information. So you might want to consider joining the forum at the earliest.**