SSN College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
Department of Computer Science and Engineering
Continuous Assessment Test – IV

Degree & Branch:	**B.E. Computer Science and Engineering**		Semester: **5**
Subject Code & Name:	**UCS1504 ARTIFICIAL INTELLIGENCE**		
Academic Year:	**2020-2021**	Batch: 2018-2022	Date: **07-11-2020** FN
Time: 120 minutes		Answer All Questions	Maximum: 50 Marks

Given n paragraphs numbered from 1 to n, arrange them in the order of $1, 2, \ldots, n$ where $n \leq 9$. With the help of a clipboard, you can press Ctrl-X (cut) and Ctrl-V (paste) several times. You cannot cut twice before pasting, but you can cut several contiguous paragraphs at the same time and these paragraphs will later be pasted in order. What is the minimum number of steps required?

Example 1 Make $[2, 4, 1, 5, 3, 6]$ sorted.

$[2, 4, 1, 5, 3, 6]$ cut paragraph (1) and paste it before paragraph (2)
$[1, 2, 4, 5, 3, 6]$ cut paragraph (3) and paste it before paragraph (4)
$[1, 2, 3, 4, 5, 6]$ Minimum number of steps required = 2

Example 2 Make $[(3, 4, 5), 1, 2]$ sorted.

$[3, 4, 5, 1, 2]$ cut three paragraphs $3, 4, 5$ at the same time and paste them after paragraph 2
$[1, 2, 3, 4, 5]$ Minimum number of steps required = 1

This solution is not unique as we can have the following alternative answer:

$[3, 4, 5, 1, 2]$ cut two paragraphs $1, 2$ at the same time and paste them before paragraph 3
$[1, 2, 3, 4, 5]$ Minimum number of steps required = 1

Example 3 A trivial algorithm will process $[5, 4, 3, 2, 1]$ as follows: $[(5), 4, 3, 2, 1] \longrightarrow [(4), 3, 2, 1, 5] \longrightarrow [(3), 2, 1, 4, 5] \longrightarrow [(2), 1, 3, 4, 5] \longrightarrow [1, 2, 3, 4, 5]$ of total 4 cut-paste steps. This is not optimal, as we can solve this instance in only 3 steps: $[5, 4, (3, 2), 1] \longrightarrow [3, (2, 5), 4, 1] \longrightarrow [3, 4, (1, 2), 5] \longrightarrow [1, 2, 3, 4, 5]$.

1. Formulate the problem as a state-space search problem. Implement a program to solve the problem using **best-first search algorithm**. You can use suitable data structures available in Python. (25)

   (a) For each function, providee code for testing your function (Program with no testing code will get lesser points).

2. Modify your program to solve the problem using **A* algorithm**. Decide a suitable heuristic for the problem. (25)

   (a) For each function, provide code for testing your function (Program with no testing code will get lesser points).

   (b) Give a measure of how much A* performs better than best-first search.