

Started on Saturday, 21 September 2024, 10:12 AM

State Finished

Completed on Saturday, 21 September 2024, 10:25 AM

Time taken 13 mins 13 secs

Grade 100.00 out of 100.00

Question 1

Correct

Mark 20.00 out of 20.00

Write a python program to implement Boyer Moore Algorithm with Good Suffix heuristic to find pattern in given text string.

For example:

Input	Result
ABAAABAACD	pattern occurs at shift = 0
ABA	pattern occurs at shift = 4

Answer: (penalty regime: 0 %)

Reset answer

```

1 def preprocess_strong_suffix(shift, bpos, pat, m):
2     ##### Add your Code here #####
3     #Start here
4     i = m
5     j = m + 1
6     bpos[i] = j
7     while i > 0:
8         while j <= m and pat[i - 1] != pat[j - 1]:
9             if shift[j] == 0:
10                 shift[j] = j - i
11                 j = bpos[j]
12                 i -= 1
13                 j -= 1
14                 bpos[i] = j
15     #End here
16 def preprocess_case2(shift, bpos, pat, m):
17     j = bpos[0]
18     for i in range(m + 1):
19         if shift[i] == 0:
20             shift[i] = j
21         if i == j:
22             j = bpos[j]

```

	Input	Expected	Got	
✓	ABAAABAACD ABA	pattern occurs at shift = 0 pattern occurs at shift = 4	pattern occurs at shift = 0 pattern occurs at shift = 4	✓
✓	SaveethaEngineering Saveetha veetha	pattern occurs at shift = 2 pattern occurs at shift = 22	pattern occurs at shift = 2 pattern occurs at shift = 22	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 2

Correct

Mark 20.00 out of 20.00

Write a Python program for Bad Character Heuristic of Boyer Moore String Matching Algorithm

For example:

Input	Result
ABAAAABCD ABC	Pattern occur at shift = 5

Answer: (penalty regime: 0 %)

Reset answer

```

1 NO_OF_CHARS = 256
2 def badCharHeuristic(string, size):
3     ##### Add your Code Here #####
4     #Start here
5     badChar = [-1]*NO_OF_CHARS
6     for i in range(size):
7         badChar[ord(string[i])] = i;
8     return badChar
9     #End here
10 def search(txt, pat):
11     m = len(pat)
12     n = len(txt)
13     badChar = badCharHeuristic(pat, m)
14     s = 0
15     while(s <= n-m):
16         j = m-1
17         while j>=0 and pat[j] == txt[s+j]:
18             j -= 1
19         if j<0:
20             print("Pattern occur at shift = {}".format(s))
21             s += (m-badChar[ord(txt[s+m])] if s+m<n else 1)
22         else:

```

	Input	Expected	Got	
✓	ABAAAABCD ABC	Pattern occur at shift = 5	Pattern occur at shift = 5	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Create a python program to find the Hamiltonian path using Depth First Search for traversing the graph .

For example:

Test	Result
hamiltonian.findCycle()	['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'A'] ['A', 'H', 'G', 'F', 'E', 'D', 'C', 'B', 'A']

Answer: (penalty regime: 0 %)

Reset answer

```

1 class Hamiltonian:
2     def __init__(self, start):
3         self.start = start
4         self.cycle = []
5         self.hasCycle = False
6
7     def findCycle(self):
8         self.cycle.append(self.start)
9         self.solve(self.start)
10
11    def solve(self, vertex):
12        ##### Add your code here #####
13        #Start here
14        if vertex == self.start and len(self.cycle) == N+1:
15            self.hasCycle = True
16            self.displayCycle()
17            return
18        for i in range(len(vertices)):
19            if adjacencyM[vertex][i] == 1 and visited[i] == 0:
20                nbr = i
21                visited[nbr] = 1
22                self.cycle.append(nbr)

```

	Test	Expected	Got	
✓	hamiltonian.findCycle()	['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'A'] ['A', 'H', 'G', 'F', 'E', 'D', 'C', 'B', 'A']	['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'A'] ['A', 'H', 'G', 'F', 'E', 'D', 'C', 'B', 'A']	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 4

Correct

Mark 20.00 out of 20.00

Write a python program to find minimum steps to reach to specific cell in minimum moves by knight.

Answer: (penalty regime: 0 %)

Reset answer

```

1 class cell:
2
3     def __init__(self, x = 0, y = 0, dist = 0):
4         self.x = x
5         self.y = y
6         self.dist = dist
7
8     def isInside(x, y, N):
9         if (x >= 1 and x <= N and
10            y >= 1 and y <= N):
11             return True
12         return False
13     def minStepToReachTarget(knightpos,
14                             targetpos, N):
15         # add your code here
16         #Start here
17         dx = [2, 2, -2, -2, 1, 1, -1, -1]
18         dy = [1, -1, 1, -1, 2, -2, 2, -2]
19         queue = []
20         queue.append(cell(knightpos[0], knightpos[1], 0))
21         visited = [[False for i in range(N + 1)] for j in range(N + 1)]
22         visited[knightpos[0]][knightpos[1]] = True

```

	Input	Expected	Got	
✓	30	20	20	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Write a Python Program to print factorial of a number recursively.

For example:

Input	Result
5	Factorial of number 5 = 120
6	Factorial of number 6 = 720

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 fact=1
3 for i in range(1,n+1):
4     fact=fact*i
5 print("Factorial of number",n,"=",fact)
6

```

	Input	Expected	Got	
✓	5	Factorial of number 5 = 120	Factorial of number 5 = 120	✓
✓	6	Factorial of number 6 = 720	Factorial of number 6 = 720	✓
✓	7	Factorial of number 7 = 5040	Factorial of number 7 = 5040	✓
✓	8	Factorial of number 8 = 40320	Factorial of number 8 = 40320	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.