

Started on Friday, 25 April 2025, 8:18 AM

State Finished

Completed on Friday, 25 April 2025, 3:31 PM

Time taken 7 hours 13 mins

Overdue 5 hours 13 mins

Grade 80.00 out of 100.00

Question **1**

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        if vertex == self.start and len(self.cycle) == N+1:
13            self.hasCycle = True
14            self.displayCycle()
15            return
16        for i in range(len(vertices)):
17            if adjacencyM[vertex][i] == 1 and visited[i] == 0:
18                nbr = i
19                visited[nbr] = 1
20                self.cycle.append(nbr)
21                self.solve(nbr)
22                visited[nbr] = 0
```

	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 **2**

Not answered

Mark 0.00 out of 20.00

Write a python to implement Quick sort using the first element as pivot value

For example:

Input	Result
5	Pivot: 61
61	Pivot: 8
24	Pivot: 24
3	Sorted array: [3, 8, 24, 50, 61]
50	
8	
6	Pivot: 2
2	Pivot: 3
3	Pivot: 54
54	Pivot: 28
10	Sorted array: [2, 3, 10, 28, 54, 94]
28	
94	

Answer: (penalty regime: 0 %)

1 ||



Question 3

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

```

	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 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 implement pattern matching on the given string using Brute Force algorithm.

For example:

Test	Input	Result
BF(a1,a2)	abcaaaabbbbccabcbabdbcsbbbbnnn ccabcba	12

Answer: (penalty regime: 0 %)

Reset answer

```

1 def BF(s1,s2):
2     i = 0
3     j = 0
4     while(i < len(s1) and j < len(s2)):
5         if(s1[i] == s2[j]):
6             i += 1
7             j += 1
8         else:
9             i = i - j + 1
10            j = 0
11    if(j >= len(s2)):
12        return i - len(s2)
13    else:
14        return 0
15
16 if __name__ == "__main__":
17     a1=input() #"abcaaaabbbbccabcbabdbcsbbbbnnn"
18     a2=input() #"ccabcba"
19     b=BF(a1,a2)
20     print(b)
21

```

	Test	Input	Expected	Got	
✓	BF(a1,a2)	abcaaaabbbbccabcbabdbcsbbbbnnn ccabcba	12	12	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.