

Practical 6

name:-Virendra Ashish Uplenchwar

Roll no:-A4_B1_05

CODE:-

```
def binaryst():
    n=4
    keys=[10,20,30,40]
    p=[0.1,0.2,0.4,0.3]
    q=[0.05,0.1,0.05,0.05,0.1]
    E=[[0]*n for _ in range(n)]
    W=[[0]*n for _ in range(n)]

    for i in range(n):
        E[i][i]=p[i]+q[i]+q[i+1]
        W[i][i]=p[i]+q[i+1]

    for length in range(2,n+1):
        for i in range(n-length+1):
            j=i+length-1
            E[i][j]=float('inf')
            W[i][j]=W[i][j-1]+p[j]+q[j+1]

            for r in range(i,j+1):
                left_cost = E[i][r-1] if r > i else 0
                right_cost = E[r+1][j] if r < j else 0
                cost = left_cost + right_cost + W[i][j]
                if cost<E[i][j]:
                    E[i][j]=cost

    return E[0][n-1]

result=binaryst()
print(result)
```

OUTPUT:-

```
2.45

=== Code Execution Successful ===
```

Task 2:-

The screenshot displays a coding platform interface with the following components:

- Top Navigation Bar:** Includes a search bar, and links for Courses, Tutorials, Practice, and Jobs.
- Left Sidebar:**
 - Output Window:** Shows 'Compilation Results' for 'Custom Input' by 'Y.O.G.J. (AI Bot)'. It confirms 'Problem Solved Successfully' with 104/104 test cases passed, 1/1 attempts, 100% accuracy, 8/8 points scored, and a time taken of 0.99.
 - Solve Next:** Lists 'Fixing Two nodes of a BST', 'Strictly Increasing Array', and 'Word Wrap'.
 - Stay Ahead With:** Promotes 'Build 21 Projects in 21 Days'.
- Main Editor:**
 - Language:** Python3.
 - Code:**

```
1 class Solution:
2     def optimalSearchTree(self, keys, freq, n):
3         E = [[0] * n for _ in range(n)]
4         W = [[0] * n for _ in range(n)]
5
6         for i in range(n):
7             E[i][i] = freq[i]
8             W[i][i] = freq[i]
9
10        for length in range(2, n + 1):
11            for i in range(n - length + 1):
12                j = i + length - 1
13                W[i][j] = W[i][j-1] + freq[j]
14                E[i][j] = float('inf')
15
16            for r in range(i, j + 1):
17                left_cost = E[i][r-1] if r > i else 0
18                right_cost = E[r+1][j] if r < j else 0
19                cost = left_cost + right_cost + W[i][j]
20
21                if cost < E[i][j]:
22                    E[i][j] = cost
23
24        return E[0][n-1]
```
 - Buttons:** 'Custom Input', 'Compile & Run', and 'Submit'.