A5-B2-29

Rishit Saxena

TASK1

def optimal\_bst(keys, p, q):

    n = len(keys)

    E = []

    for i in range(n + 1):

        row = []

        for j in range(n + 1):

            row.append(0.0)

        E.append(row)

    W = []

    for i in range(n + 1):

        row = []

        for j in range(n + 1):

            row.append(0.0)

        W.append(row)

    R = []

    for i in range(n + 1):

        row = []

        for j in range(n + 1):

            row.append(0)

        R.append(row)

    for i in range(n + 1):

        E[i][i] = q[i]

        W[i][i] = q[i]

    for d in range(1, n + 1):

        for i in range(n - d + 1):

            j = i + d

            E[i][j] = float('inf')

            W[i][j] = W[i][j - 1] + p[j - 1] + q[j]

            for k in range(i + 1, j + 1):

                cost = E[i][k - 1] + E[k][j] + W[i][j]

                if cost < E[i][j]:

                    E[i][j] = cost

                    R[i][j] = k

    return E[0][n]

print("Enter number of keys:")

n = int(input())

print("Enter the keys :")

keys\_input = input().split()

keys = []

for val in keys\_input:

    keys.append(int(val))

print("Enter probabilities of keys (p):")

p\_input = input().split()

p = []

for val in p\_input:

    p.append(float(val))

print("Enter probabilities of dummy keys (q) :")

q\_input = input().split()

q = []

for val in q\_input:

    q.append(float(val))

result = optimal\_bst(keys, p, q)

print("\nMinimum Expected Search Cost:", format(result, ".4f"))



