```
#include <bits/stdc++.h>
using namespace std;

// Function to calculate sum of frequencies from index i to j
int sum(int freq[], int i, int j) {
   int s = 0;
   for (int k = i; k <= j; k++)
      s += freq[k];
   return s;
}

// Recursive function to find the minimum cost of Optimal BST
int optCost(int freq[], int i, int j) {
   if (j < i)
      return 0;
   if (i == j)
      return freq[i];

   int fsum = sum(freq, i, j);
   int minCost = INT_MAX;

   // Try making all keys in interval [i, j] the root
   for (int r = i; r <= j; ++r) {
      int cost = optCost(freq, i, r - 1) + optCost(freq, r + 1, j);
      if (cost < minCost)
         minCost = cost;
   }

   return minCost + fsum;
}
```

```
// Function to find the cost of the Optimal BST
int optimalSearchTree(int keys[], int freq[], int n) {
   return optCost(freq, 0, n - 1);
}


int main() {
   int number_of_keys;
   cout << "\nEnter number of keys: ";
   cin >> number_of_keys;


   int keys[number_of_keys], freq[number_of_keys];
   cout << "\n";


   for (int i = 0; i < number_of_keys; ++i) {
      cout << "Enter key and its frequency: ";
      cin >> keys[i] >> freq[i];
   }


   cout << "\nCost of Optimal BST: " << optimalSearchTree(keys, freq, number_of_keys) << "\n";


   return 0;
}
```