```
1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Linq;
5  using System.Text;
6
7  namespace Algorithms
8  {
9      /// <summary>
10     /// page 297
11     /// </summary>
12     public class Binary_Search_Tree
13     {
14         public static void Main()
15         {
16             new Binary_Search_Tree().Run(Console.In, Console.Out);
17         }
18         void Run(TextReader reader, TextWriter writer)
19         {
20             writer.Write(OptimalBST(new int[] { 1, 3, 4, 6, 5, 7, 8, 9, 10 }));
21             reader.Read();
22         }
23         /// <summary>
24         /// Finds an optimal binary search tree by dynamic programming
25         /// </summary>
26         /// <param name="P">An array P[1..n] of search probabilities for a sorted
               list of n keys</param>
27         /// <returns>Average number of comparisons in successful seraches in the
               optimal BST and table R of subtress roots in the Optimal BST</returns>
28         Tuple<int, int[,]> OptimalBST(int[] P)
29         {
30             int[,] C = new int[P.Length, P.Length];
31             int[,] R = new int[P.Length, P.Length];
32             for (int i = 1; i < P.Length; i++)
33             {
34                 C[i, i - 1] = 0;
35                 C[i, i] = P[i];
36                 R[i, i] = i;
37             }
38             C[P.Length + 1, P.Length] = 0;
39             for (int d = 0; d < P.Length - 1; d++)//diagonal count
40             {
41                 for (int i = 1; i < P.Length - d; i++)
42                 {
43                     int j = i + d;
44                     int minval = int.MaxValue;
45                     for (int k = i; k <= j; k++)
46                     {
47                         int kmin = 0;
48                         if (C[i, k - 1] + C[k + 1, j] < minval)
49                         {
50                             minval = C[i, k - 1] + C[k + 1, j];
51                             kmin = k;
52                         }
53                         R[i, j] = kmin;
```

```
54                          int sum = P[i];
55                          for (int s = i + 1; s <= j; s++)
56                          {
57                              sum = sum + P[s];
58                          }
59                          C[i, j] = minval + sum;
60                      }
61                  }
62              }
63              return Tuple.Create(C[1, P.Length], R);
64          }
65      }
66  }
67
```