

## 动态规划：873. 最长的斐波那契子序列的长度



解题思路：

优化点：将所有元素和下标放在hash中，以O1的时间复杂度快速找到对应的下标，避免了循环。

$dp[i][j]$  表示以i位置j位置为下标元素的，最长严格递增斐波那契子序列的长度

代码：

```

1  class Solution {
2  public:
3      int lenLongestFibSubseq(vector<int>& arr) {
4          int n = arr.size();
5          int max1 = 2;
6          unordered_map<int,int> hash;
7          for(int i = 0 ;i < n ;i++)
8          {
9              hash[arr[i]] = i;
10         }
11         vector<vector<int>> dp(n,vector<int>(n,2));
12
13         for(int j = 2 ;j < n;j++)
14         {
15             for(int i = 1;i < j;i++)
16             {
17                 int a = arr[j] - arr[i];
18                 if(hash.count(a) && a < arr[i])
19                 {
20                     dp[i][j] = dp[hash[a]][i] + 1;
21                 }
22                 max1 = max(max1,dp[i][j]);
23             }
24         }
25         return max1 < 3 ? 0 :max1;
26     }
27 };

```