

## 动态规划: [1218. 最长定差子序列](#)

### 解题思路:

💡 这个题目很有意思，解题的时候用上了hash，帮助我们快速索引对应差的数值，我个人学习到了，如何在hash (unorderedmap) 中快速索引，并且修改对应value值

### 代码:

```

1  class Solution {
2  public:
3      int longestSubsequence(vector<int>& arr, int difference) {
4
5          // int n = arr.size();
6          // int max1 = 0;
7          // unordered_map<int, int> hash;
8
9          // for (int i = 0; i < n; i++) {
10         //     // 检查当前元素前一个元素是否存在于哈希表中
11         //     if (hash.find(arr[i] - difference) == hash.end()) {
12         //         hash[arr[i]] = 1; // 如果不存在, 则以1初始化
13         //     } else {
14         //         hash[arr[i]] =
15         //             hash[arr[i] - difference] + 1; // 如果存在, 更新为之前的值+1
16         //     }
17         //     max1 = max(max1, hash[arr[i]]); // 更新最大子序列长度
18         // }
19         // return max1;
20
21         int n = arr.size();
22         int max1 = 0;
23         unordered_map<int, int> hash;
24         for(int i = 0 ;i < n; i++)
25         {
26             if(hash.find(arr[i] - difference) == hash.end())
27                 //if(hash[arr[i] - difference].count() == 0)
28                 {
29                     hash[arr[i]] = 1;
30                     //hash.insert(make_pair([arr[i] - difference],1);
31                 }
32             else
33             {
34                 hash[arr[i]] = hash[arr[i] - difference] + 1;
35                 //hash.insert(make_pair([arr[i]],hash[arr[i] -
36                 //difference]+1);
37             }
38             max1 = max(max1,hash[arr[i]]);

```

```
39     }
40     return max1;
41
42     // int n = arr.size();
43     // int max1 = 0;
44     // vector<int> dp(n,1);
45     // for(int i = 0; i < n;i++)
46     // {
47     //     for(int j = i ;j >=0 ;j--)
48     //     {
49     //         if(arr[i] - arr[j] == difference)
50     //         {
51     //             dp[i] = dp[j] + 1;
52     //             break;
53     //         }
54     //     }
55     //     max1 = max(max1,dp[i]);
56     // }
57     // return max1;
58 }
59 };
```