1218. 最长定差子序列

给你一个整数数组 arr 和一个整数 difference，请你找出并返回 arr 中最长等差子序列的长度，该子序列中相邻元素之间的差等于 difference 。

子序列 是指在不改变其余元素顺序的情况下，通过删除一些元素或不删除任何元素而从 arr 派生出来的序列。

Prolog

```
输入：arr = [1,2,3,4], difference = 1
输出：4
解释：最长的等差子序列是 [1,2,3,4]。

输入：arr = [1,3,5,7], difference = 1
输出：1
解释：最长的等差子序列是任意单个元素。

输入：arr = [1,5,7,8,5,3,4,2,1], difference = -2
输出：4
解释：最长的等差子序列是 [7,5,3,1]。
```

动态规划：这个题目比较容易理解，在遍历到当前元素时计算上一个与自己相差difference值的元素，那个元素是否存在，如果存在就拿它的值+1就获得当前元素以自己为最后一个元素的最长子序列；与此同时，记录最大元素最后返回。

这里为什么不用dp数组呢，数组有下标，map存放的是key，这里需要比较的是和上一个与自己相差difference的值所对应的最长子序列，不是前一个元素，所以这里不用数组。

时间与空间复杂度都是O(N)

Java

```java
class Solution {
    public int longestSubsequence(int[] arr, int difference) {
        int ans = 0;
        Map<Integer, Integer> dp = new HashMap<>();
        for(int v : arr) {
            // 存放当前元素的最长子序列，并记录最大值
            dp.put(v, dp.getOrDefault(v - difference, 0) + 1);
            ans = Math.max(ans, dp.get(v));
        }
        return ans;
    }
}
```