

青蛙过河

题目：

一只青蛙想要过河。假定河流被等分为 x 个单元格，并且在每一个单元格内都有可能放有一石子（也有可能没有）。青蛙可以跳上石头，但是不可以跳入水中。

给定石子的位置列表（用单元格序号升序表示），请判定青蛙能否成功过河（即能否在最后一步跳至最后一个石子上）。开始时，青蛙默认已站在第一个石子上，并可以假定它第一步只能跳跃一个单位（即只能从单元格1跳至单元格2）。

如果青蛙上一步跳跃了 k 个单位，那么它接下来的跳跃距离只能选择为 $k - 1$ 、 k 或 $k + 1$ 个单位。另请注意，青蛙只能向前方（终点的方向）跳跃。

请注意：

- 石子的数量 ≥ 2 且 < 1100 ；
- 每一个石子的位置序号都是一个非负整数，且其 $< 2^{31}$ ；
- 第一个石子的位置永远是0。

示例 1：

[0,1,3,5,6,8,12,17]

总共有8个石子。

第一个石子处于序号为0的单元格的位置，第二个石子处于序号为1的单元格的位置，

第三个石子在序号为3的单元格的位置，以此定义整个数组...

最后一个石子处于序号为17的单元格的位置。

返回 **true**。即青蛙可以成功过河，按照如下方案跳跃：

跳1个单位到第2块石子，然后跳2个单位到第3块石子，接着

跳2个单位到第4块石子，然后跳3个单位到第6块石子，

跳4个单位到第7块石子，最后，跳5个单位到第8个石子（即最后一块石子）。

示例 2：

[0,1,2,3,4,8,9,11]

返回 **false**。青蛙没有办法过河。

这是因为第5和第6个石子之间的间距太大，没有可选的方案供青蛙跳跃过去。

思路：

设置dp为跳转到第j块石头的最后一步跳的距离i时能否成功。

cur-stone[i]：从第i块石头跳转到当前跳了need步，

根据题意，如果need-1，need+1和need在这个dp内，则认为可以跳转过去，如果不在，则不能跳转过去。

```
1 def canCross(self, stones: List[int]) -> bool:
2     n = len(stones)
3     %23 跳转到第i个石头的最后一步距离是j时能否成功
```

```
4         %23 这里采用的时候hash表, set()
5         dp = [set() for _ in range(n)]
6         %23 初始值, 第0块石头, 跳转了0步
7         dp[0].add(0)
10        for i in range(n):
11            %23 当前跳转到第i块石头
12            cur = stones[i]
13            for j in range(i):
14                %23从第j块石头跳转到第i块石头的步数,
15                need = cur - stones[j]
16                %23 根据题意, 上一步跳跃k步, 接下来只能跳跃k, k+1, k-1, 所以, 判
断need-1, need, need+1, 是否在上一步跳跃的次数中, 是, 就说明跳跃成功, 就把此次跳跃
加入
17                if need - 1 in dp[j] or need + 1 in dp[j] or need in dp[j]
:
18                    dp[i].add(need)
19
20        return len(dp[-1]) > 0
```