

思路

我们定义一个概念:“覆盖范围”,还没有跳跃之前,可以到达的最远的地方

定义: `curDistance`, 就是还没有跳远之前,可以到达的最远的地方; `nextDistance`, 就是: 当前跳跃了一次后,可以到达的最远的地方

根据题意,我们只是要求最少的步骤,我们实际上并不要求“上一次跳跃停下来的地方”,因此我们就不用记录“上一次跳跃停下来的地方”

贪心的策略就是: 期望到达的最远的地方

base case: 初始情况下, `curDistance=0`, `nextDistance = arr[0] + 1`

`i`从下标0开始,遍历数组,如果`i`没有在当前的覆盖范围内了,也就是`curDistance == i`了,那么就进行一次跳跃,重新赋值`curDistance=nextDistance`

`i`从上一次跳跃停下的地方,到`curDistance`这段遍历的时期,我们不断的更新`nextDistance=max`,这里使用了贪心策略,我们想要获得尽可能大的`nextDistance`