Method 1: dynamic programming.

State transition equation:

$$f(i) = \max(\underbrace{f(i-1)}_{\text{max. jumps leftover from previous state}}, \underbrace{nums[i-1]}_{\text{no. of jumps from previous position}}) - \overbrace{1}^{\text{to reach current position need one jump}}$$

input:

$nums = [2, 3, 1, 1, 4]$

$f(0) = 0$

$f(1) = \max(0, 2) - 1 = 1$

$f(2) = \max(1, 3) - 1 = 2$

$f(3) = \max(2, 1) - 1 = 1$

$f(4) = \max(1, 1) - 1 = 0$

Path: 2 - 3 - 4

Method 2: greedy algorithm.

input:

$nums = [2, 3, 1, 1, 4]$

$leftmost = 4$

$i = 5 - 2 = 3$; $3 + 1 >= 4$, $leftmost = 3$

$i = 2$: $2 + 1 >= 3$, $leftmost = 2$

$i = 1$: $1 + 3 >= 2$, $leftmost = 1$

$i = 0$: $0 + 2 >= 1$, $leftmost = 0$