COMP 305 | Quiz 2
Minimum Jumps

Gul Sena Altintas - 64284

May 21, 2021

Contents

1 Problem

Given an array of n elements, where $arr[i]$ represents the maximum number of jumps we can move from position i, find the minimum number of jumps to reach the end of the array.

Assumptions: If $arr[i] = 0$, we cannot move forward from i, if $arr[0] = 0$, return -1.

2 Solution

Algorithm 1 Minimum Jumps

1: $jumps \leftarrow$ array of n elements initialized with ∞
2: **for** $i \leftarrow n-1$ to 0 **do**
3: Calculate $jumps[i]$ with Eq 1

We can formulate a dynamic programming approach as follows:

$$jumps[i] = \begin{cases} 1 & \text{if arr[i] + i} \geq n \\ -1 & \text{if arr[i]=0} \\ 1 + \min_j jumps[j] & \text{otherwise, where } j \in \{i+1 \dots \min\{i+1+arr[i], n\}\} \text{ and } arr[j] > 0 \end{cases} \quad (1)$$

In Eq. 1 $jumps[i]$ represents the minimum number of jumps to reach $arr[n-1]$ from position i. Hence, we start filling $jumps$ from $n-1$, and the solution is given by $jumps[0]$.

Note that in the actual implementation, we can choose to store ∞ or -1 in the $jumps$ array. I chose to store -1 rather than ∞ in my implementation.

3 Analysis

The total number of subproblems -filling the dynamic programming table, ie. array $jumps$- is $O(n)$. Each subproblem is dominated by the min operation which in the worst case is $O(n)$. Hence, the overall algorithm has time complexity $O(n^2)$ and $O(n)$ space complexity to store array $jumps$.