Solution to an Example Monotonic Path Problem

Let A be an n by n matrix where each entry $A[i][j]$ is a positive integer. We say an entry $A[i][j]$ is a z-entry if $A[i][j] \equiv z \pmod{3}$. We say a path *monotonic* if the path goes only upward and rightward. If a path visits c_0 0-entries, c_1 1-entries, and c_2 2-entries, then the path scores $2c_0 + 3c_1 + c_2$ points. Give an $O(n^2)$-time algorithm to determine what is the highest points that a monotonic path from $A[1][1]$ to $A[n][n]$ can score. Give the pseudocode of your algorithm and explain why it runs in $O(n^2)$ time.

Solution:

Let $\alpha(P)$ denote the points that path P scores. Let $sol[x][y]$ denote $\max_P \alpha(P)$ among all monotonic paths from $A[1][1]$ to $A[x][y]$. Hence, $A[n][n]$ gives the solution. The pseudocode of our algorithm is given as follows, and the initial call is $\textsc{Find}(n, n, sol = \{-\infty\})$.

if $sol[x][y] > -\infty$ **then**
    return $sol[x][y]$;
end
$count \leftarrow (A[x][y] \bmod 3) + 2$;
if $count > 4$ **then**
    $count \leftarrow 1$;
end
if (x, y) *equals* $(1, 1)$ **then**
    return $sol[x][y] = count$;
end
if x *equals* 1 **then**
    return $sol[x][y] = count + \textsc{Find}(x, y - 1, sol)$;
end
if y *equals* 1 **then**
    return $sol[x][y] = count + \textsc{Find}(x - 1, y, sol)$;
end
return $sol[x][y] = count + \max\{\textsc{find}(x - 1, y, sol), \textsc{find}(x, y - 1, sol)\}$;

Algorithm 1: $\textsc{Find}(x, y, sol)$

It takes $O(1)$ time to fill in each entry in sol, and there are $O(n^2)$ entries in sol. The total running time is thus $O(n^2)$.