Be a Dynamo!

March 14, 2017

Assume that you have an array of integers C with indexes $1 \ldots n$.
Consider the following recurrence:

$$A(i) = \begin{cases} 1 & \text{if } i < 5 \text{ or } i > n - 5 \\ \min(C[i-1] + A(i-1), C[i-2] + A(i-2)) & \text{if } 5 \leq i < \lfloor \frac{n}{2} \rfloor \\ \min(C[i-1] + A(i-1), C[i+1] + A(i+1)) & \text{if } i = \lfloor \frac{n}{2} \rfloor \\ \min(C[i+1] + A(i+1), C[i+2] + A(i+2)) & \text{otherwise} \end{cases}$$

Convert this to an efficient dynamic programming solution that computes $A(\lfloor \frac{n}{2} \rfloor)$. Your solution should take C (and n if desired) as a parameter. Feel free to assume you have a helper function `IsPow2(x)` that returns true if x is a power of 2.