Lab 9 Continued

We can certainly use Dynamic Programming. The easiest way (for me) is to run the recurrence from Day 0 up to Day n-1, with a recurrence defined like this:

Let Z(d,j) be the maximum number of zombies we can fry, starting on Day d, given that the ZZ has been charging for j days.

Let A(d) be the number of zombies arriving on Day d.

On Day d, with j days of charging, we only have 2 choices: fire the ZZ or not. If we fire it, we fry min{A(d),32,2^(j-1)} zombies, and we recurse on Day d+1 with the days of charging reduced to 1. If we don't fire the ZZ on Day d, we fry 0 zombies and recurse on Day d+1 with the days of charging increased to j+1.

Notice that we should always fire the ZZ on the last day (Day n-1) – there is no reason to save the charge.

This gives us the recurrence relation:

$Z(n-1,j) = \min\{A(n-1),32,2^{(j-1)}\}$ for all values of j

$$Z(d,j) = \max \left\{ \begin{array}{l} \min\{A(d),32,2^{(j-1)}\} + Z(d+1,1), \\ Z(d+1,j+1) \end{array} \right\}$$ for all values of j, and all $d < n-1$

It is not clear to me that we can use a Greedy approach on this problem. Choosing to fire the ZZ on the day with the most zombies arriving is definitely not going to find the optimal solution. There may be some way to prioritize the days and then choose days to fire the ZZ from the prioritized list ... but I don't see it.