(a) Suppose that $n = 6$ and the coordinates are $3, 2, -3, -2, -1, 0$. Then the greedy algorithm would observe events $2, 5, 6$, while an optimal solution would observe events $3, 4, 5, 6$.

(b) Let $OPT(j)$ denote the maximum number of events that can be observed, subject to the constraint that event j is observed. Note that $OPT(n)$ is the value that we want.

To define a recurrence for $OPT(j)$, we consider the previous event before j that is observed in an optimal solution. If it is i, then we need to have $|d_j - d_i| \leq j - i$, and we behave optimally up through observing event i. Thus we have

$$OPT(j) = 1 + \min_{i:|d_j - d_i| \leq j-i} OPT(i).$$

The values of OPT can be built up in order of increasing j, in time $O(j)$ for iteration j, leading to a total running time of $O(n^2)$. The value we want is $OPT(n)$, and the configuration can be found by tracing back through the array of OPT values.

[1] ex259.807.630