Subtask 1: Don't wait

Obviously, it doesn't make sense for Tycho to wait for p or more seconds at the starting position. So we can simply try all possible times at $t \in \{0, 1, \ldots, p-1\}$ at which Tycho starts moving. It remains to compute the damage. Tycho takes $b + t$ units of environmental damage. For the radiation damage, we need to count the number of pulses at which Tycho is not sheltered. When the pulses happen, Tycho will be at $p - t, 2p - t, \ldots$. There are $\left\lceil \frac{b-(p-t)}{p} \right\rceil$ such positions which are $< b$. From those, we subtract the number of $a_i \equiv p - t \pmod{p}$.

Subtask 2: Bruteforce

There is always an optimal solution, which waits only at sheltered positions and the waited time is until the next pulse. We can simply try all 2^n subsets of shelters to wait there and calculate the damage function.

Subtask 3: $b \leq 1000$

Let $dp[x][t]$ be the minimum damage, when Tycho is at position x with a time $\equiv t \pmod{p}$. Tycho has two options: wait or move to the next position. This gives us the new position $x' \in \{x, x+1\}$ and the new time $t' = t + 1 \bmod p$. The cost of this transition is 1 and additional d if $t' = 0$ and x' is not sheltered.
Notice however, that the resulting "DP graph" is cyclic. To solve this problem, we can either use Dijkstra to find the shortest path or observe that there is an optimal solution where we don't visit $(x, 0)$ from $(x, p-1)$. Instead, we could simply visit it from $(x-1, p-1)$. So we can remove the transition $(x, p-1) \rightarrow (x, 0)$ and the graph becomes acyclic again. The DP can now be calculated in $\mathcal{O}(pb)$.

Subtask 4: A quadratic DP

Suppose that we already chose the shelters where Tycho waits – like in Subtask 2. The other shelters in between can be ignored. So Tycho wants to move between two selected shelters x, y without waiting. It takes damage at positions $x+p, x+2p, \ldots$ which are $< y$. There are $\left\lceil \frac{y-x}{p} \right\rceil - 1$ such positions. And since Tycho waits at position y, the environmental damage is $p \left\lceil \frac{y-x}{p} \right\rceil$. The total damage only depends on the length $l = y - x$ and is given by:

$$f(l) = (d + p) \left\lceil \frac{l}{p} \right\rceil - d$$

This way, we can solve the problem with dynamic programming: Let $dp[i]$ be the minimum damage for Tycho to reach shelter a_i at a time divisible by p.

$$dp[i] = \min_{j<i} (dp[j] + f(a_i - a_j))$$

which can be calculated in $\mathcal{O}(n^2)$. The last part where Tycho moves to the base has to be calculated separately, because Tycho does not need to wait at the base:

$$ans = \min_i (dp[i] + g(b - a_i))$$
$$g(l) = l + d\left(\left\lceil \frac{l}{p} \right\rceil - 1\right)$$

Subtask 5: Small p

Look at the DP transition above. Not all j are relevant. Indeed, we only need to consider the maximum $j < i$ for which $a_j \equiv x \pmod p$ for some x. This reduces the complexity to $\mathcal{O}(np)$. There are at most n different remainders, so it can also be implemented in $\mathcal{O}(n \min(n, p))$.

Full solution

Look at the DP formula again and let $a_i = q_i p + r_i$ and $a_j = q_j p + r_j$ with $0 \le r_i, r_j < p$. Then,

$$\left\lceil \frac{a_i - a_j}{p} \right\rceil = q_i - q_j + \left\lceil \frac{r_i - r_j}{p} \right\rceil = q_i - q_j + [r_i > r_j]$$

Let's handle the cases $r_j < r_i$ and $r_j \ge r_i$ separately. In the first case, the transition looks like this:

$$dp[i] = \min_{\substack{j<i \\ r_j<r_i}} (dp[j] + (d+p)(q_i - q_j + 1) - d) = \min_{\substack{j<i \\ r_j<r_i}} (dp[j] - (d+p)q_j) + (d+p)(q_i + 1) - d$$

For the second case we get:

$$dp[i] = \min_{\substack{j<i \\ r_j \ge r_i}} (dp[j] - (d+p)q_j) + (d+p)q_i - d$$

Thus, we can put all $dp[j] - (d+p)q_j$ in an RMQ data structure at position $a_j \bmod p$. And to compute a_i, we make two queries: One for the prefix $< r_i$ and one for the suffix $\ge r_i$. For example, we can use a segment tree for this which solves Subtask 6 in $\mathcal{O}(p + n\log(p))$. However, p is too big for the last subtask. This can be solved via coordinate compression or an implicit segment tree.