(a) We prove this by induction on $d$. If $d = 0$, then $\Phi$ is a satisfying assignment, and $\textit{Explore}(\Phi, d)$ returns “yes.”

Now consider $d > 0$. If $\textit{Explore}(\Phi, d)$ returns “yes,” it is because one of the recursive calls $\textit{Explore}(\Phi_i, d - 1)$ returns “yes”; by induction, this means that $\Phi_i$ has distance $d - 1$ to a satisfying assignment, and so $\Phi$ has distance $d$ to a satisfying assignment.

Conversely, suppose $\Phi$ has distance $d$ to a satisfying assignment $\Phi'$. Consider any clause unsatisfied by $\Phi$; since $\Phi'$ satisfies it, it must disagree with $\Phi$ on the setting of at least one of the variables in this clause. Thus, one of the assignments $\Phi_i$, which changes the assignment to this variable, is at distance $d - 1$ to $\Phi'$; by induction the recursive call $\textit{Explore}(\Phi_i, d - 1)$ will return “yes,” and so the full call $\textit{Explore}(\Phi, d)$ will also return “yes.”

The running time for $\textit{Explore}$ satisfies the recurrence $T(n, d) \leq 3T(n, d - 1) + p(n)$, for a polynomial $p$. Unwinding this to get $d$ down to 0, we have a running time of $O(3^d \cdot p(n))$.

(b) We let $\Phi_0$ denote the assignment in which all variables are set to 0, and we let $\Phi_1$ denote the assignment in which all variables are set to 1. If there is any satisfying assignment, it is within distance at most $n/2$ of one of these, so we can call both $\textit{Explore}(\Phi_0, n/2)$ and $\textit{Explore}(\Phi_1, n/2)$, and see if either of these returns “yes.”

The running time of each of these calls is $O(p(n) \cdot 3^{n/2}) = O(p(n) \cdot (\sqrt{3})^n)$.

\footnote{ex695.88.327}