(a) We prove this by induction on d. If $d = 0$, then Φ is a satisfying assignment, and $Explore(\Phi, d)$ returns "yes."

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

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

The running time for *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 Φ_0 denote the assignment in which all variables are set to 0, and we let Φ_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 $Explore(\Phi_0, n/2)$ and $Explore(\Phi_1, n/2)$, and see if either of these returns "yes."

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

[1] ex695.88.327