Solution

a) We can prove that $Explore(\Phi, d)$ works properly by induction on d. If d is 0, and Φ is a satisfying argument then the algorithm returns yes, so it works. Now we look at the problem where $d > 0$. If $Explore$ returns yes, then one of the recursive calls to $Explore(\Phi_i, d-1)$ must have returned yes. So that would mean there is a solution to Φ_i within $d-1$ of a satisfying argument, which means that there is a solution to Φ within a distance d. Because we're breaking up the problem into 3 subproblems each time, the running time of this algorithm satisfies the recurrence $T(n,d) = 3T(n, d-1) + p(n)$. If we solve this recurrence, we get a running time of $O(p(n) * ((\sqrt{(}3))^n)$.

b) We can do this by considering two assignments Φ_0 and Φ_1 where the former sets all variables to 0, and the latter sets them all to 1. Now we observe that any satisfying assignment is within distance $n/2$ of either Φ_0 or Φ_1, rather than n. So we solve the problem by calling $Explore(\Phi_1, n/2)$ and $Explore(\Phi_0, n/2)$ and returning yes if either of these calls return yes. This solution takes time $O(p(n) * 3^{n/2})$ so it has run time of $O(p(n) * 3^n)$