HW7

Daniel Ginsburg

1.

In this equation $a = 1$, $b = 2$, and $d = 0$. Thus, $a > b^d$, and therefore the O(n) is $= n^{log_b(a)} = n^{log_2(2)} = n^1 = n$.

2.

Let n be the position in the array containing the last valid integer.

First you look for an ∞. You do this by looking at the 2^{ith} integer as i increases starting from 0, until you hit an ∞. You know that $i \leq log_2(n) + 1$. Thus the highest possible position you will check is around arr[2n], and it will be done in O(log(n)) time. Once you find the ∞, you do a binary search (with 1 and 2n as the start and end) to find the x (if it is in the array). You assume ∞ is bigger than x. This operation is also $O(log(2n)) = O(log(n))$ because $log_2(2n) = log_2(n) + 1$. Because both operations are O(log(n)), the total run time is O(log(n)).