For Bob's algorithm, he used the divide-n-conquer strategy, and this algorithm has time efficiency $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + \Theta(n)$. And for convenience, we'll omit the floors, ceilings, and boundary conditions, and hence,

$$T(n) = T(n/2) + T(n/2) + \Theta(n) = 2T(n/2) + \Theta(n).$$

By using the master method, for this recurrence, we have a = 2, b = 2, f(n) = n, and $n^{\log_b a} = n^{\log_2 2} = n^1 = n$. Case 2 applies, since $f(n) = \Theta\left(n^{\log_b a}\right) = \Theta(n)$, and thus the solution to the recurrence is $T(n) = \Theta\left(n^{\log_b a} \lg n\right) = \Theta(n\, lg\, n)$.

While for Alice's algorithm, she used the incremental strategy, and this algorithm has time efficiency $T(n) = \begin{cases} \Theta(1) & if\ i = 1, \\ T(n-1) + \Theta(c) & otherwise \end{cases}$, where c is some constant, and i is the number the algorithm takes in as parameter. So, we adopt c in place of Θ(c), and 1 for Θ(1).

T(n) → T(n – 1) + c

→ T((n – 1) – 1) + c + c

→ T(1) + c + … + c

→ 1 + (n – 1)c

= Θ(n)

Hence, in Θ(·) notation, Alice's algorithm takes linear time in the worst case, and her algorithm is more time-efficient than Bob's.