Colors

Subtask 1 ($N \leq 64$)

We will use the colors in this order: 1, N, 2, $N-1$, 3, $N-2$, …; this way we will check each difference $N-1$, $N-2$, $N-3$, $N-4$, … and the answer is the first difference that is not recognized by Archie.

Complexity: N queries.

Subtask 2 ($N \leq 125$)

We can first ask the colors $N/2$ and 1. If Archie recognizes the difference, then $C \leq N/2$, and as in the Subtask 1, we can ask the queries $N/2-1$, 2, $N/2-2$, 3, $N/2-3$, 4, …, until we find the first difference that Archie does not recognize. Otherwise we ask the queries N, 2, $N-1$, 3, $N-2$, … until we find the first difference that Archie recognizes.

Complexity: $N/2+1$ queries.

Subtask 3 ($N \leq 1000$)

First use $\sqrt{N}$ values and try to understand k value for which it is true that C is between $k\sqrt{(n)}$ and $(k+1)\sqrt{(n)}$. For example, if $N = 100$, then use values 5, 15, 25, …, 95 and use the Subtask 1 N-query algorithm to find the value of k. And then again use the Subtask 1 N-query algorithm to calculate the precise C value.

Complexity: $2\sqrt{N}$ queries.

Subtask 4 ($N \leq 10^9$)

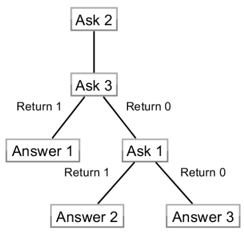
Let assume that we have a correct strategy for all values of N that do not exceed k.

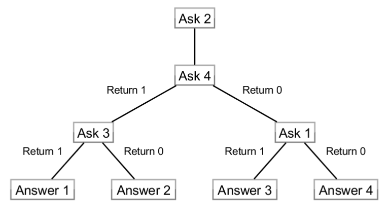
If k is even ($k = 2j$) we will use the strategy that was used for j numbers and use only even (or odd) numbers. This way each jump in j becomes twice as long and in the result (when the strategy for j has finished) we will know that the answer is 1 or 2, 3 or 4, 5 or 6, and so on. We then know for some x that Archie recognizes the difference $2x$ and we need to understand whether he recognizes the difference $2x-1$. It can be proved that if possible answers are $2x-1$ and $2x$, then the last difference that was checked was either $2x$ or $2x-2$ and in both cases we will be able to make a jump in the opposite direction with length $2x-1$.

If k is odd ($k = 2j+1$) we use the strategy for j colors and use the numbers 2, 4, 6, 8, 10, and so on. When the strategy for j is finished, we know that the answer is 1 or 2, 3 or 4, 5 or 6, …, $2j-1$ or $2j$ or $2j+1$. And then in almost all cases we can calculate the answer with one additional query, but if the possible answer is one of $2j-1$, $2j$ or $2j+1$ then we need to use two additional queries.

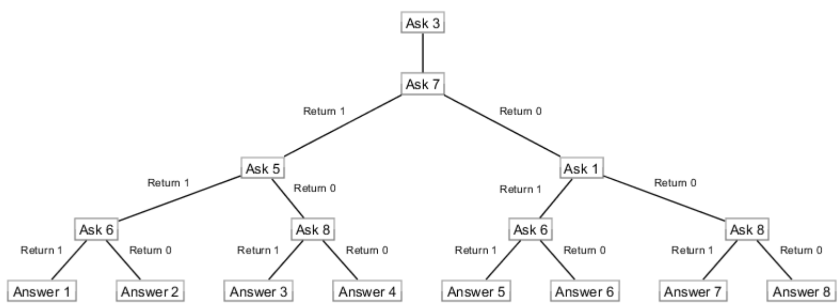
Complexity: $2\log_2 N$ queries.

For example, for the base cases $N = 3$ and $N = 4$ we can use the following algorithms:





Then using our construction, we get the following algorithm for $N = 8$:



Subtask 5 ($N \leq 10^{18}$)

Let's assume that we have a correct strategy for all values of N that do not exceed k. We will restrict our strategy even more – consecutive jumps need to be made in the opposite directions.

Suppose that k is even ($k = 2j$) and the first color used in the j strategy is f. Then we make the first jump from f to $f + j$ (or from $f + j$ to f). With this jump we will understand whether C is bigger than j (if the answer is negative) or smaller or equal than j (if the answer is positive). If the answer is smaller or equal to j then we use strategy for j on numbers from 1 to j (we already have the color f). If the answer is bigger than j, then we extend all jumps in the j strategy by j (if we had a jump with length p, then now we will make jump with length $p+j$). As we are always jumping back and forth then we will never jump out of range from 1 to n and will return the answer in the range $j+1$ to $2j$.

If k is odd, we can use a similar strategy.

Complexity: $\log_2 N + 1$ queries.

In this case, we get the following algorithm for $N = 8$: