

```

oneByOne(n)
{
    m = 0;
    while (n != 1)
    {
        if (n % 23 == 0) n /= 23;
        else { n++; m++; }
    }
    return m;
}

```

Demonstrăm că algoritmul se oprește inductiv.

$P(k)$ - algoritmul se oprește pentru valoarea $n=k$
(echivalent, când valoarea n la intrarea în while este $n=k$)

$n=1$: $P(1)$: alg se oprește pt $n=1$
este "A" deoarece nu intră în bucla while.

Pp. că alg se oprește pt orice $n \in \{1, \dots, k-1\}$ și
dăm că se oprește pt $n=k$.

pentru $n=k \neq 1$ programul intră în bucla while
- dacă $k \% 23 = 0$, n este modificat: $n = n / 23$
 \Rightarrow noua valoare a lui $n \in \{1, \dots, k-1\}$
pentru care știm din ipoteza inductivă
că programul se oprește.

- dacă $k \% 23 \neq 0$, $n = n + 1$
dar această instrucțiune este repetată
până când $n \% 23 = 0$ adică de
 $23 - (k \% 23)$ ori după care
noua valoare a lui n va fi
 $n = k + (23 - k \% 23)$ pentru
care $n \% 23 = 0$ este "A" și se
execută instrucțiunea $n = n / 23$:

Acum argumentăm că noua valoare
a lui n este în $\{1, \dots, k-1\}$

$$m = \left(k + (23 - k \% 23) \right) / 23 < k \Leftrightarrow$$

$$\Leftrightarrow k + (23 - k \% 23) < 23 * k \Leftrightarrow$$

$$\Leftrightarrow 23 - k \% 23 < 22 * k \Leftrightarrow$$

$$\Leftrightarrow 23 < 22k + k \% 23 \quad \textcircled{1}$$

Dar $k > 1$ și $k \% 23 \neq 0$ deci

$$k > 1 \text{ și } k \% 23 \geq 1 \Rightarrow$$

$$\Rightarrow 22k > 22 \quad \Bigg| \Rightarrow 22k + k \% 23 > 23 \Rightarrow$$

\Rightarrow relația $\textcircled{1}$ este "A" \Rightarrow

\Rightarrow noua valoare a lui n este în
mulțimea $\{1, \dots, k-1\}$ pentru care
știm că alg. se oprește.

Cazul cel mai nefavorabil:

$$n = 23 + 1 = 24 \Rightarrow 24 + \underbrace{1 + 1 + \dots + 1}_{22 \text{ ori}} = 46 / 23 = 2$$

$$2 + \underbrace{1 + \dots + 1}_{21 \text{ ori}} = 23 / 23 = \boxed{1} \text{ stop.}$$

$$(n + 22) / 23 = 23 \cdot k + 1 \Rightarrow n + 22 = 23(k+1) + 23 \Rightarrow$$

$$\Rightarrow n = 23(k+1) + 1$$