

Jako první můj algoritmus zkontroluje, zda $N = 1$. Pokud ano, vrátí rovnou 1. Pak tehdy začne samotný algoritmus.

Pro fungování tohoto algoritmu musíme nejprve pro zadané N zkonstruovat finite-state machine, který bude přijímat jazyk složený z čísel z nul a jedniček, které jsou dělitelné N . Víme, že počet stavů tohoto stroje je roven počtu zbytkových množin. Hrany grafu tohoto stroje pak určíme snadno.

Nechť x je stav, ze kterého vede hrana, a y_0 a y_1 jsou stavy, kde končí hrany, pokud je aktuální číslice 0 nebo 1. Pokud budeme ta čísla kontrolovat tak, že číslice budeme zadávat od začátku, pak y_0 je rovno zbytku po přidání nuly na konec čísla a y_1 je rovno zbytku po přidání jedničky. Pak tedy platí:

$$y_0 \equiv 10 \cdot x \pmod{N}$$

$$y_1 \equiv 10 \cdot x + 1 \pmod{N}$$

To nám stačí znát ke konstrukci finite-state machine přijímající jazyk čísel z nul a jedniček dělitelné N .

Když už tedy máme zkonstruovaný finite-state machine pro dané N , nejmenší číslo dělitelné N pak získáme tím, že nalezneme nejkratší cestu v grafu tohoto stroje. Pokud se nalezne více nejkratších cest, pak vybereme tu, která má nulu na nejvyšší pozici. Toho můžeme dosáhnout tak, že spustíme prohledávání do šířky ze stavu se zbytkem 1 do stavu se zbytkem 0. Do fronty pak budeme vždy přidávat nejdříve hrany, kdy se přidává nula, a pak ty hrany, kdy se přidává jednička. Následně po prohledávání vyvodíme z cesty, kterou jsme prošli, výsledek, nebo prohlásíme, že dané číslo neexistuje. Časová a prostorová složitost tohoto algoritmu je pak $\mathcal{O}(N)$, protože počet stavů a počet hran je přímo úměrný číslu N a prohledávání do šířky trvá lineárně vůči počtu vrcholů a hran.