

**Příklad 1.**

Naivní algoritmus, který zkouší všechny možné začátky jehly v seně a vždy porovnává řetězce, má časovou složitost  $\mathcal{O}(JS)$ . Může být opravdu tak pomalý, uvážíme-li, že porovnávání řetězců skončí, jakmile najde první neshodu? Sestrojte vstup, na kterém algoritmus poběží  $\Theta(JS)$  kroků, přestože nic nenajde.

**Definice.**

Máme-li řetězec  $\alpha$ , nazvěme jeho *rotaci* o  $k$  pozic řetězec  $\alpha[k:] \alpha[:k]$ .

**Příklad 2.**

Jak o dvou řetězcích zjistit, zda je jeden rotací druhého?

**Příklad 3.**

Dostali jsme slovo. Jak najít jeho nejdelší vlastní prefix, který je současně jeho suffixem?

**Příklad 4.**

Sestavte KMP automaty pro následující řetězce:

- KOCKA
- AAAAAAAAAA
- TORONTO
- KOCKODAN