Jako první přiřadíme ke každému jedinečnému řádku podmatice, kterou hledáme, jedinečný symbol. K tomu využijeme trie, a to tak, že postupně budeme každý řádek do něj přidávat a v případě, kdy se daný řádek ještě v trii nenachází, přiřadíme řádku nový znak, jinak tento řádek označíme znakem, který odpovídá dané sekvenci znaků. Toto potrvá $\mathcal{O}(K^2)$ času a může zabrat až $\mathcal{O}(K^2)$ prostoru. Každý jedinečný řádek podmatice pak bude tvořit jehelníček, a tedy můžeme využít algoritmus Aho-Corasickova k tomu, abychom v každém řádku matice našli všechny sekvence písmen, které odpovídají nějakému řádku podmatice. Při hledání těchto sekvencí si při nalezení nějaké jehly vyznačíme konec této jehly právě pomocí symbolu, který jsme danému řádku přiřadili na začátku. To si můžeme dovolit, protože všechny řádky jsou stejně dlouhé a tím pádem se nemůže stát, že bychom měli dva nálezy v jeden okamžik. Tuto část algoritmu stihneme v čase $\mathcal{O}(N^2 + K^2)$ a spotřebuje $\mathcal{O}(N^2)$ prostoru na ukládání konců nalezených jehel.

Po části výše budeme mít v matici vyznačeny místa, kde končí sekvence, které mohou odpovídat nějakému řádku podmatice. Toho teď využijeme k tomu, abychom zjistili, zda se zadaná podmatice ve velké matici nachází. To, co k tomu musíme udělat, je prohledávat sloupce a hledat takovou sekvenci znaků, která odpovídá znakům označující řádky hledané podmatice přesně v tom pořadí, jako tato podmatice má. Na toto můžeme znova použít algoritmus Aho-Corasickova nebo si vystačíme s KMP algoritmem, jelikož hledáme jen jedinou jehly. Tato část potrvá $\mathcal{O}(N^2 + K)$ času.

Celkově tento algoritmus potrvá $\mathcal{O}(N^2 + K^2)$ času a zabere stejně prostoru, tedy prostorová složitost je $\mathcal{O}(N^2 + K^2)$.