1. *Pěstírna stromů:* Pěstovaný strom říkejme zakořeněnému stromu, jehož hrany k synům mají v každém vrcholu určené uspořádaní. Strom se osekává tak, že si vybereme kořen podstromu, vše mimo podstrom odstraníme a pak ještě můžeme odseknout některé hrany zleva a zprava v kořeni (zbude tedy souvislý úsek hran z kořene podstromu dolů a podstromy, které pod nimi visí). Jak zjistit o dvou pěstovaných stromech, zda lze jeden získat osekáním druhého?

2. *Velikost výstupu:* Ukažte, že velikost výstupu (počet dvojic (k, i) takových, že na pozici k začíná i-té slovo) může být velká (superlineární $\omega(n)$, kde n je součet velikostí sena a jehel).

3. *Seznamy nefungují:* Proč si nelze pro každý stav s pamatovat množinu slov $M(s)$, jejichž výskyty chceme hlásit?

4. *Naivní hledání končících slov:* Ukažte, že skákání po zpětných hranách trvá dlouho.

5. *Slovník:* Pro slovník vytvořte datovou struktru, která odpovídá na dotazy typu "je ve slovníku rotace slova X"?

6. *Rýmovník:* Pro slovník si vybudujte datovou strukturu, která umí k zadanému slovu efektivně najít nejlepší rým (s nejdelším společným suffixem).

7. *Frekvenční analýza:* Pro dané podřetězce spočtěte jejich četnosti v textu.

8. *Nejčastější podslovo:* Chceme hledat nejčastěji se vyskytující podřetězec délky k.

---

9. *Cenzor:* Cenzor dostane text a množinu zakázaných řetězců. Cenzor nalezne nejlevější výskyt zakázaného řetězce ten smaže a postup opakuje dokud existuje zakázaný vzor v textu. Navrhněte algoritmus, který usnadní cenzorovu práci.

10. *Zašifrované seno:* Hledáme každý možný výskyt jehly v seně zašifrovaném substituční šifrou (seno má zpermutovanou abecedu).

11. *Lexikograficky minimální:* Najděte lexikograficky minimální rotaci.

12. *Dynamické vyhledávání:* Navrhněte datovou strukturu pro dynamické vyhledávání v textu. Jehla je pevná, v seně lze průběžně měnit jednotlivé znaky a struktura odpovídá, zda se v seně právě vyskytuje jehla.

1. *Hlášení vybraných výskytů (6 bodů):* Vymyslete algoritmy, které dostanou na vstupu slova $s_1, s_2, \ldots, s_k$ a text T. Výstupem algoritmu pak je pro každou pozici,
   (a) na které **končí** nějaké slovo: index pozice a **nejdelší** slovo, které na ní **končí**,
   (b) na které **končí** nějaké slovo: index pozice a **nejkratší** slovo, které na ní **končí**,
   (c) na které **začíná** nějaké slovo: index pozice a **nejdelší** slovo, které na ní **začíná**,
   (d) na které **začíná** nějaké slovo: index pozice a **nejkratší** slovo, které na ní **začíná**.

Nezapomeňte, že se hodnotí i srozumitelnost řešnení a že správné řešení obsahuje:

1. slovně dostatečně detailně popsaný algoritmus (volitelně pseudokód),
2. důkaz správnosti,
3. časovou složitost se zdůvodněním,
4. prostorovou složitost se zdůvodněním.