1. *Opakování:* KMP - kokos a clanekokokosu
2. *Naivní hledání:* Dokažte, že naivní algoritmus pro vyhledávání v textu (zkouším všechny pozice a pro každou porovnám s celým hledaným řetězcem) může běžet až $\Omega(JS)$ kroků, kde J je délka hledaného řetězce (jehly) a S délka sena, a to i tehdy, když vůbec nic nenajde.
3. *Rotace:* Jak zjistit, jestli je jeden řetězec rotací druhého?
4. *Periodicita:* Jak zjistit, jestli je zadaný řetězec periodický?
5. *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?
6. *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).

---

7. *Pěstírna stromů:* viz výše
8. *Zašifrované seno:* Hledáme každý možný výskyt jehly v seně zašifrovaném substituční šifrou (seno má zpermutovanou abecedu).
9. *Lexikograficky minimální:* Najděte lexikograficky minimální rotaci.
10. *Frekvenční analýza:* Pro dané podřetězce spočtěte jejich četnosti v textu.
11. *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.
12. *Rýmovník:* Pro slovník si vybudujte datovou strukturu, která umí k zadanému slovu efektivně najít nejlepší rým (s nejdelším společným suffixem).

1. *Suffixy a prefixy (5 bodů):* Pro slova a, b chceme nalézet nejdelší suffix slova a, který je zároveň prefixem slova b.