1. *Nejčastější výskyt:* Zjistěte jaké slovo délky k se vyskytuje nejčastěji jako podslovo slova σ?
2. *Nejdelší opakovaný podřetězec:* Najděte nejdelší podslovo σ, které se v σ vyskytuje alespoň dvakrát.
3. *Vícero zdrojů/stoků:* Jak najít maximální tok v situaci, kdy mám zdrojů a stoků více?
4. *F-F s celými čísly:* Jak rychle doběhne Ford-Fulkersonův algoritmus pro jednotkové váhy a jak rychle (pokud vůbec) pro celočíselné?
5. *Špatná síť:* Najděte příklad malé sítě, na níž může Ford-Fulkersonův algoritmus provést více než milion iterací.
6. *Hranově disjunktní cesty:* Navrhněte algoritmus pro nalezení maximálního počtu hranově disjunktních cest mezi danými dvěma vrcholy $u, v \in V(G)$.
7. *Vrcholově disjunktní cesty:* Navrhněte algoritmus pro nalezení maximálního počtu vrcholově disjunktních cest mezi danými dvěma vrcholy $u, v \in V(G)$.

8. *Nejdelší společný:* Ukažte, jak pro dané dvě slova najít jejich nejdelší společné podslovo.

9. *Nejdelší palindrom:* Najděte nejdelší palindromické podslovo slova σ. Existují poměrně složité deterministické lineární algoritmy, nicméně s pomocí hashování byste měli být schopní dosáhnout složitosti alespoň $O(n \log n)$.

10. *Nekonečný F-F:* Najděte síť s reálnými kapacitami, na níž Fordův-Fulkersonův algoritmus nedoběhne. Lze zařídit, aby k maximálnímu toku ani nekonvergoval?

11. *Nečekaná záchrana:* Přímočará implementace Fordova-Fulkersonova algoritmu bude nejspíš graf prohledávat do šířky, takže vždy najde nejkratší nenasycenou cestu. Pak překvapivě platí, že algoritmus zlepší tok jen $\mathcal{O}(nm)$-krát, než se zastaví. Návod k důkazu: Nechť $\ell(u)$ je vzdálenost ze zdroje do vrcholu u po nenasycených hranách. Nejprve si rozmyslete, že $\ell(u)$ během výpočtu nikdy neklesá. Pak dokažte, že mezi dvěma nasyceními libovolné hrany uv se musí $\ell(u)$ zvýšit. Proto každou hranu nasytíme nejvýše $\mathcal{O}(n)$-krát.