1. *Goldberg a výšky:* Co by se stalo, kdybychom v inicializaci Goldbergova algoritmu umístili zdroj do výšky $n-1$, $n-2$, anebo dokonce $n-3$? Rozmyslete si, která vlastnost (skončí, vydá vždy maximální tok, ...) na výšce zdroje závisí a tedy, která by se mohla pokazit.

2. *Goldberg a jednotkové sítě:* Rozeberte chování Goldbergova algoritmu na sítích s jednotkovými kapacitami. Bude rychlejší než ostatní algoritmy? Nebo alespoň stejně rychlý?

3. *Věže na šachovnici:* Mějme šachovnici $r \times s$, z níž políčkožrout sežral některá políčka. Chceme na ni rozestavět co nejvíce šachových věží tak, aby se navzájem neohrožovaly. Věž můžeme postavit na libovolné nesežrané políčko a ohrožuje všechny věže v témže řádku i sloupci. Navrhněte efektivní algoritmus, který takové rozestavění najde.

4. *Věže podruhé:* Stejně jako předchozí příklad, ale tentokrát věž „nevidí" přes sežraná políčka.

5. *Parlamentní kluby:* V parlamentu s n poslanci je m různých klubů. Jeden poslanec může být členem mnoha různých klubů. Každý klub nyní potřebuje zvolit svého předsedu a tajemníka tak, aby všichni předsedové a tajemníci byli navzájem různé osoby (tedy aby nikdo „neseděl na více křeslech"). Navrhněte algoritmus, který zvolí všechny předsedy a tajemníky, případně oznámí, že řešení neexistuje. Mimochodem, za jakých podmínek je existence řešení garantována?

6. *Průchod šachovnicí:* Je dána šachovnice $n \times n$, kde některá políčka jsou nepřístupná. Celý dolní řádek je obsazen figurkami, které se mohou hýbat o jedno pole dopředu, šikmo vlevo dopředu, či šikmo vpravo dopředu. V jednom tahu se všechny figurky naráz pohnou (mohou i zůstat stát na místě), na jednom políčku se však musí vyskytovat nejvýše jedna figurka. Ocitne-li se figurka na některém políčku horního řádku šachovnice, zmizí. Navrhněte algoritmus, který najde minimální počet tahů takový, že z šachovnice dokážeme odstranit všechny figurky, případně oznámí, že řešení neexistuje.

7. *Minimální izolace:* Je dán nepříliš velký celočíselný kvádr (dejme tomu s objemem nejvýše 20 000) a v něm k nebezpečných jednotkových kostiček. Navrhněte algoritmus, který najde podmnožinu M kostiček kvádru takovou, že každá nebezpečná kostička leží v M a zároveň M má minimální možný povrch (povrch měříme jako počet takových stěn kostiček z M, které nesousedí s jinou kostičkou z M).

8. *Rychlejší Goldberg:* Navrhněte implementaci vylepšeného Goldbergova algoritmu se zvedáním nejvyššího vrcholu s přebytkem. Snažte se dosáhnout časové složitosti $\mathcal{O}(n^2\sqrt{m})$.