1. *Omezené kapacity:* Dokažte, že pro jednotkové kapacity Dinicův algoritmus doběhne v čase $\mathcal{O}(nm)$. Dokažte totéž pro celočíselné kapacity omezené konstantou.

2. *Hranově disjunktní cesty:* Navrhněte algoritmus pro nalezení maximálního počtu hranově disjunktních cest mezi danými dvěma vrcholy $u, v \in V(G)$.

3. *Vrcholově disjunktní cesty:* Navrhněte algoritmus pro nalezení maximálního počtu vrcholově disjunktních cest mezi danými dvěma vrcholy $u, v \in V(G)$.

4. *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.

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

6. *Dopravní problém:* Uvažujme továrny $T_1, \ldots, T_p$ a obchody $O_1, \ldots, O_q$. Všichni vyrábějí a prodávají tentýž druh zboží. Továrna T_i ho denně vyprodukuje t_i kusů, obchod O_j denně spotřebuje o_j kusů. Navíc známe bipartitní graf určující, která továrna může dodávat zboží kterému obchodu. Najděte efektivní algoritmus, který zjistí, zda je požadavky obchodů možné splnit, aniž by se překročily výrobní kapacity továren, a pokud je to možné, vypíše, ze které továrny se má přepravit kolik zboží do kterého obchodu.

7. *Zaokrouhlování matice:* Na vstupu dostaneme matici A nezáporných reálných čísel o velikosti $r \times s$. Vymyslete algoritmus, který zaokrouhlí prvky matice nahoru nebo dolů tak, že zůstanou zachovány všechny řádkové i sloupcové součty, nebo odpoví, že takové zaokrouhlení neexistuje.

8. *k-souvislost:* Díky 2. cvičení umíme pro dva vrcholy u, v zjistit, zda mezi nimi vede k hranově disjunktních cest. Pomocí tohoto algoritmu zjistěte zda je neorientovaný graf G hranově k-souvislý. (Graf je hranově k-souvislý pokud mezi každými dvěma vrcholy vede k hranově disjunktních cest.) Váš algoritmus nemá dostatek času na to, aby zavolal $\Theta(n^2)$-krát algoritmus z 2. cvičení.

9. *Doly a továrny:* Uvažujeme o vybudování dolů $D_1, \ldots, D_p$ a továren $T_1, \ldots, T_q$. Vybudování dolu D_i stojí cenu d_i a od té doby důl zadarmo produkuje neomezené množství i-té suroviny. Továrna T_j potřebuje ke své činnosti zadanou množinu surovin a pokud jsou v provozu všechny doly produkující tyto suroviny, vyděláme na továrně zisk t_j. Vymyslete algoritmus, jenž pro zadané ceny dolů, zisky továren a bipartitní graf závislostí továren na surovinách stanoví, které doly postavit, abychom vydělali co nejvíce.

10. *Algoritmus tří Indů:* Blokující tok ve vrstevnaté síti lze nalézt chytřejším způsobem v čase $\mathcal{O}(n^2)$, čímž zrychlíme celý Dinicův algoritmus na $\mathcal{O}(n^3)$. Následuje stručný popis, doplňte k němu detaily.

    Pro každý vrchol v definujme $r^+(v)$ jako součet rezerv na všech hranách vedoucích do v. Nechť dále $r^-(v)$ je totéž přes hrany vedoucí z v a $r(v) = \min(r^+(v), r^-(v))$ „rezerva vrcholu". Pokud je $r(v)$ všude 0, tok už je blokující.

    V opačném případě opakovaně vybíráme nejmenší $r(v)$ a snažíme se ho vynulovat. Potřebujeme tedy dopravit $r(v)$ jednotek toku ze zdroje do v a totéž množství z v do stoku. Popišme dopravu do stoku (ze zdroje postupujeme symetricky): ve vrcholech udržujeme plán $p(w)$, který říká, kolik potřebujeme z w dopravit do stoku. Na začátku je $p(v) = r(v)$ a všechna ostatní $p(w) = 0$. Procházíme po vrstvách od v ke stoku a pokaždé plán převedeme po hranách s kladnou rezervou do vrcholů v další vrstvě. Jelikož $r(v) \leq r(w)$ pro všechna w, vždy nám to vyjde. Průběžně čistíme slepé uličky.