

Příklad 1.

Ukažte, jak v Dinicově algoritmu naprogramovat hledání blokujícího toku a dodatečného čištění sítě rezerv jako (opakované) prohledávání do hloubky.

Příklad 2.

Dokažte, že máme-li v síti celočíselné kapacity omezené konstantou, Dinicův algoritmus doběhne v čase $\mathcal{O}(nm)$.

Příklad 3.

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?

Příklad 4.

Na louce je n svišťů a m děr v zemi (zadáno jako body v rovině). Když se objeví orel, zvládne svišť uběhnout pouze d metrů, než bude uloven. Kolik nejvýše svišťů se může zachránit útekem do díry, když jedna díra je dost velká jen pro jednoho sviště? A co když se do díry vejde k svišťů?

Příklad 5.

Graf je hranově k -souvislý, jestliže je souvislý a po odebrání libovolných $k - 1$ hran bude stále souvislý. Najděte algoritmus, který zjistí pro jaké největší k je daný graf hranově k -souvislý. Může se vám hodit hledání toků v až $\mathcal{O}(n)$ sítích.

Příklad 6.

Uvažujme továrny T_1, \dots, T_p a obchody O_1, \dots, O_q . Všechny továrny vyrábějí stejný druh zboží. Továrna T_i za den vyrobí t_i kusů, obchod O_j 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.

***Příklad 7.**

Je dána šachovnice $k \times k$, 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. Ocitněli 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.

Můžete nejprve určit horní odhad počtu tahů.