

Za domácí úkol máte příklady 5 a 6. Důkladně si přečtěte zadání!

U všech níže uvedených příkladů se snažte najít algoritmus, který zadanou úlohu vyřeší co možná nejrychleji s co nejmenší spotřebou paměti. Vždy uvádějte časovou a paměťovou složitost vašich algoritmů i s důkazy správnosti vašich tvrzení.

Příklad 1. Uvažujme vektorový grafický editor, který umí pracovat s různými geometrickými objekty (např. body, úsečky, elipsy, beziérky). Editor umí vytvářet závislosti pozic objektů mezi sebou. Pozice objektů mohou být vstupní (např. pevné souřadnice bodu nebo aktuální pozice myši) nebo záviset na pozicích jiných objektů (např. pozice bodu může být dána průsečíkem dvou úseček). Pozice objektu může záviset na několika objektech (např. kružnice může být dána třemi body). Pokud uživatel změní pozici vstupních objektů nebo závislosti mezi objekty, tak je nutné přepočítat pozice všech objektů ke správnému překreslení. Najděte algoritmus, který určí, v jakém pořadí se pozice objektů měla přepočítávat.

Příklad 2. Mějme souvislý neorientovaný graf G . V jakém pořadí odtrhávat vrcholy tak, aby graf zůstal souvislý? Tj. najděte algoritmus, který najde posloupnost všech vrcholů v_1, \dots, v_n grafu G takovou, že $G \setminus \{v_1, \dots, v_k\}$ je souvislý pro všechna $k = 1, \dots, n - 1$.

Příklad 3. Mějme neorientovaný graf G a počáteční vrchol s . Vymyslete algoritmus, který pro každý vrchol v najde nejen délku nejkratší cesty z s do v , ale i počet nejkratších cest. Dvě cesty z s do v považujeme za různé, pokud se liší v alespoň jednom vrcholu.

Příklad 4. Najděte algoritmus, který rozhodne, zda daný graf obsahuje cyklus.

Příklad 5. Mějme bludiště zadané grafem. Víme, ve kterém vrcholu se nachází princezna a ve kterém se nachází vchod. Dále víme, ve kterých chodbách stojí drak, kterého je nutné zabít, abychom mohli projít. Najděte cestu k princezně, po které je nutné zabít co nejmenší počet draků (je jich málo, a tak jsou chráněni).

Příklad 6. Mějme bludiště zadané grafem. Víme, ve kterém vrcholu se nachází princezna a ve kterém se nachází vchod. Dále pro každou hranu známe vrchol, ve kterém se nachází klíč odemykající danou hranu. Najděte algoritmus, který rozhodne, zda je možné princeznu vysvobodit.

Příklad 7. Máme šachovnici velikosti $n \times n$. Jak najít nejmenší počet tahů koně, kterými přesuneme z daného počátečního místa do daného cílového místa.

Příklad 8. Navrhněte algoritmus, který rozhodne, zda zadaný graf je bipartitní. Tak se říká grafům, jejichž vrcholy lze rozdělit na dvě množiny tak, aby koncové vrcholy každé hrany patřily do různých množin.