1. Vyjádřete výrazem pomocí výsledků DFS vlastnost „u je předkem v“ pro vrcholy zakořeněného stromu.

   $\mathrm{in}(u) < \mathrm{in}(v) \wedge \mathrm{out}(v) < \mathrm{out}(u)$.

2. Zjistěte pomocí výsledků DFS, jestli je graf bipartitní.

   Podívám se na zakořeněný strom průchodu DFS a vezmu $\mathrm{lvl}(v)$ jako číslo úrovně, na které leží vrchol v. Graf je bipartitní ⇔ každý cyklus je sudý. To platí právě když každá zpětná hrana uv překlenuje lichý počet úrovní ($\mathrm{lvl}(u) - \mathrm{lvl}(v) \bmod 2 = 1$).

3. Jaká je časová složitost DFS na různě reprezentovaných grafech (alespoň maticí sousednosti a seznamem sousedů)? Jak to dopadne pro husté grafy (řekněme $m \in \Theta(n^2)$)?

   DFS na grafu reprezentovaném maticí sousedností seběhne v čase $\mathcal{O}(n^2)$ a na grafu reprezentovaném seznamem sousedů v čase $\mathcal{O}(n + m)$. Pokud $m \in \Theta(n^2)$, tak oba vyjdou stejně.

4. DFS na orientovaném grafu rozdělí hrany na *stromové*, *dopředné*, *příčné* a *zpětné*. Které z nich mohou/musí být/nemohou být součástí orientovaného cyklu?
   Jak je to pro neorientovaný graf?

   Součástí cyklu mohou být všechny hrany, zpětně hrany jsou součástí cyklu vždy.

5. Vyjádřete vlastnost „hrana uv je součástí nějakého cyklu“, pak pomocí něj najděte algoritmus rozhodující, jestli je daný graf hranově 2-souvislý.

   Již víme, že zpětné hrany jsou vždy součástí cyklu. U stromových hran si můžeme všimnout, že jsou součástí cyklu právě tehdy, když je „překlenuje“ zpětná hrana - z podstromu spodního vrcholu vede zpětná hrana do horního nebo výše.

   Označme si jako $\mathrm{low}(v)$ nejmenší $\mathrm{in}(u)$ pro všechny u horní vrcholy zpětných hran z podstromu v (tedy $\mathrm{low}(v) =$

$\min\{\mathrm{in}(u) \mid uw \text{ zpětná hrana}, w \text{ potomek } v\}$). Intuitivně „jak vysoko se jednou zpětnou hranou dostaneme z tohoto podstromu“.

Hrana uv (s u otevřeným v DFS před v) je součástí nějakého cyklu právě když $\mathrm{low}(v) \leq \mathrm{in}(u)$.

Graf je hranově 2-souvislý právě když nemá žádné mosty. V lineárním čase to ověříme tak, že při průchodu DFS postupně budeme přepočítávat low a pak ověříme podmínku odvozenou výše.

6. Spočítejte počet všech cest mezi dvěma vrcholy u a v.

Projít všechny možné cesty můžeme jednoduše prohledáváním grafu do hloubky z u. Bohužel asymptoticky lepší algoritmus neexistuje, nějakého zrychlení by šlo pro řidší grafy dosáhnout kontrakcí delších cest, kdy pak nemusíme trávit čas procházením míst bez větvení.

7. Spočítejte počet *nejdelších* cest mezi dvěma vrcholy u a v v lineárním čase. Porovnejte přístup přes DFS a BFS (prohledávání do šířky).

Projdeme graf z u pomocí BFS a rozdělíme vrcholy na „vrstvy“ podle vzdálenosti od u. Z grafu pak smažeme všechny vrstvy za v a smažeme hrany bez cílového vrcholu. U každého zbylého vrcholu w si pak budeme pamatovat $s(w)$ - kolik nejkratších cest do něj vede z u.

$s(u)$ nastavíme na 1. Pak projdeme graf po vrstvách, v každé vrstvě projdeme každou hranu xy a nastavíme $s(y) \leftarrow s(y) + s(x)$. Výsledný počet všech nejkratších cest najdeme v $s(v)$.

Jelikož průchod přes DFS nezkonstruuje vrstevnatý graf, nemůžeme nejkratší cesty hledat takto přímočaře. Ve vrstevnatém grafu algoritmus funguje, protože každá hrana každé nejkratší cesty jde z vrstvy i do vrstvy $i + 1$, což pro hrany DFS stomu neplatí.

8. Najděte korespondenci mezi topologickým uspořádáním a výsledky DFS průchodu DAGem.

Topologické uspořádání odpovídá otočenému pořadí $\mathrm{out}(v)$ z DFS.
Pro každou (orientovanou) hranu uv (krom zpětné) platí $\mathrm{out}(u) > \mathrm{out}(v)$ a graf se zpětnými hranami nemá topologické uspořádání.

9. Charakterizujte grafy, které mají právě jedno topologické uspořádání.

Graf G má právě jedno topologické uspořádání $\Leftrightarrow$ G má Hamiltonovskou cestu.

$\Rightarrow$ G je nutně souvislý, jinak by šla topologická uspořádání dílčích komponent libovolně uspořádat. Pokud má v topologickém uspořádání $v_1, v_2, ..., v_n$ každý vrchol v_i hranu do vrcholu v_{i+1}, máme Hamiltonovskou cestu. Jinak můžeme v topologickém uspořádání v_i a v_{i+1} prohodit, protože tím neporušíme invariant směru pro žádnou hranu.

$\Leftarrow$ Vezmeme pořadí vrcholů na cestě jako topologické uspořádání. Kdyby existovala hrana „zprava doleva", dostali bychom cyklus, protože z každého v_i vede cesta do každého v_j pro $j > i$.

