1. Máte černou skříňku, která říká, jestli má daná formule splňující ohodnocení. Jak nějaké takové splňující ohodnocení naleznete? Černou skříňku můžete použít vícekrát.

2. Máte algoritmus pro problém Nezávislá množina, tj. černou skřínku která na vstup (G, k) odpoví jestli graf G obsahuje nezávislou množinu velikosti k. Jak pomocí polynomiálně mnoha volání tohoto algoritmu nalezneme maximální nezávislou množinu v grafu?

3. Navrhněte polynomiální algoritmus pro problém Nezávislá množina pokud je vstupem strom.

4. Popište jak byste řešili Součet podmnožiny pomocí dynamického programování. Proč to není ve sporu s NP-úplností tohoto problému?

5. Hledejme vrcholové pokrytí následujícím hladovým algoritmem. V každém kroku vybereme vrchol nejvyššího stupně, přidáme ho do pokrytí a odstraníme ho z grafu i se všemi již pokrytými hranami. Je nalezené pokrytí nejmenší? Nebo alespoň $\mathcal{O}(1)$-aproximace nejmenšího? Hint: zkuste zkonstruovat bipartitní graf, kde popsaný algoritmus vybere jako pokrytí větší z partit.

6. Uvažujme následující algoritmus pro nejmenší vrcholové pokrytí grafu. Graf projdeme do hloubky, do výstupu vložíme všechny vrcholy vzniklého DFS stromu kromě listů. Dokažte, že vznikne vrcholové pokrytí a že 2-aproximuje to nejmenší.

7. V daném orientovaném grafu hledáme acyklický podgraf s co nejvíce hranami. Navrhněte polynomiální 2-aproximační algoritmus.

8. Řešení problému obchodního cestujícího hrubou silou by prohledávalo graf do hloubky a zkoušelo všechny hamiltonovské kružnice. To může v grafu na n vrcholech trvat až $n!$ kroků. Pokuste se najít rychlejší algoritmus. Dynamickým programováním lze dosáhnout složitosti $\mathcal{O}(2^n \cdot n^k)$ pro konstantní k. To je sice exponenciální, ale stále mnohem lepší než faktoriál.