1. *Neunikátní Jarník:* Fungoval by Jarníkův algoritmus pro neunikátní váhy hran?
2. *Neunikátní Borůvka:* Fungoval by Borůvkův algoritmus pro neunikátní váhy hran?
3. *Neunikátní Kruskal:* Fungoval by Kruskalův algoritmus pro neunikátní váhy hran?
4. *Změna hrany:* Máte nalezenou minimální kostru. Jak najít novou minimální kostru pokud:
   (a) Z grafu odstraníme jednu hranu.
   (b) Zvýšíme váhu jedné hrany.
   (c) Snížíme váhu jedné hrany.
5. *L-kostry:* Vymyslete algoritmus na hledání kostry grafu, v němž jsou váhy hran přirozená čísla z množiny $\{1, \ldots, L\}$.

6. *Druhá nejmenší kostra:* Známe-li minimální kostru, jak najít druhou nejmenší?

7. *Kontrahující Borůvka:* Borůvkův algoritmus můžeme upravit, aby každý strom lesa udržoval zkontrahovaný do jednoho vrcholu. Iterace pak vypadá tak, že si každý vrchol vybere nejlehčí incidentní hranu, tyto hrany zkontrahujeme a zapamatujeme si, že patří do minimální kostry. Ukažte, jak tento algoritmus implementovat tak, aby běžel v čase $O(m \log n)$. Jak si poradit s násobnými hranami a smyčkami, které vznikají při kontrakci?

8. *Rovinný Borůvka:* Když šikovně použijeme algoritmus z předchozího cvičení pro rovinné grafy jak rychle najdeme minimální kostru? Opět je potřeba správně ošetřit násobné hrany.

9. *Všechny kostry:* Rozmyslete si, jak v případě, kdy váhy nejsou unikátní, najít všechny minimální kostry. Jelikož koster může být mnoho (pro úplný graf s jednotkovými vahami jich je n^{n-2}), snažte se o co nejlepší složitost v závislosti na velikosti grafu a počtu minimálních koster .