Minimální kostra grafu G: Podgraf H t.ž. H je strom obsahující každý vrchol G a součet vah jeho hran je minimální.

Jarníkův algoritmus: „Postupně budujeme kostru přidáváním nejlehčí sousední hrany."
Borůvkův algoritmus: „Udržujeme les a v každém kroku každý strom sloučíme s nejbližším sousedem."
Kruskalův algoritmus: „Postupně budujeme les koster přidáváním hran od nejlehčí po nejtěžší. Vynecháme hrany, které by přidaly cyklus."

Union-Find: Datová struktura udržující dynamicky komponenty souvislosti.
Rozhraní:
- $\text{Find}(u, v)$ - Zjistí, jestli u a v leží ve stejné komponentě souvislosti.
- $\text{Union}(u, v)$ - Přidá hranu $\{u, v\}$, tedy spojí komponenty, ve kterých se vrcholy nacházejí.

1. Najděte algoritmus, který v lineárním čase najde minimální kostru grafu s jednotkově ohodnocenými hranami.
2. Co se stane v Jarníkově/Borůvkově/Kruskalově algoritmu, když váhy hran nebudou unikátní?
   Jak můžeme algoritmy upravit, aby i v takovém případě našly unikátní minimální kostru?
3. Stručně popište konkrétní implementaci tří algoritmů na hledání minimální kostry, určete jejich asymptotickou složitost a seřaďte je podle časové složitosti.
4. Najděte co nejrychlejší algoritmus pro hledání minimální kostry v grafu s hranami ohodnocenými přirozenými čísly $1, 2, ..., k$.
4. Popište jednoduchou Union-Find datovou strukturu, která obě operace zvládne v čase $O(n)$.
5. Popište Union-Find datovou strukturu, která pomocí udržování lesa na vrcholech původního grafu operace zvládne v čase $O(\log n)$.
6. Dokažte, že hrana je most právě tehdy, když je součástí každé kostry.

