

**Semestrální práce**  
Největší klika v neorientovaném grafu  
IAL

Lukáš Lev, 256660

November 24, 2024

# 1 Zadání

Náhradní projekt je určen pouze pro studenty, kteří v předmětu IFJ neřeší souběžný projekt (např. studenti FEKT nebo studenti opakující předmět). Tento projekt je týmový a řeší jej trojice nebo čtveřice studentů.

**Zadání varianty** *Klika grafu* je podgraf, který je úplným grafem (=kterýkoliv vrchol kliky je tedy spojen hranou se všemi ostatními vrcholy kliky).

Vytvořte program pro hledání největší kliky v neorientovaném grafu. Pokud existuje více řešení, naleznete všechna. Výsledky prezentujte vhodným způsobem. Součástí projektu bude načítání grafů ze souboru a vhodné testovací grafy. V dokumentaci uveďte teoretickou složitost úlohy a porovnejte ji s experimentálními výsledky.

## Všeobecné informace a pokyny k náhradním projektům

Řešení bude vypracováno v jazyce C a bude přeložitelné (pomocí příkazu `make`) na serveru `eva.fit.vutbr.cz`. Všechny zdrojové kódy, hlavičkové soubory, testovací data aj. budou logicky separovány a uloženy v příhodně pojmenovaných podadresářích. Použití nestandardních knihoven není dovoleno. Všechny části zadání varianty jsou nutnou součástí řešení.

Celkové hodnocení projektu sestává z následujících kategorií:

- funkčnost implementace (až 6 bodů),
- projektová dokumentace (až 4 body),
- obhajoba (až 5 bodů).

Řešení zabalené v jediném ZIP archivu je odevzdáváno pouze vedoucím týmu prostřednictvím STUDISu. Závazné pokyny pro vypracování projektové dokumentace a doporučení pro závěrečné obhajoby naleznete v Moodle v sekci Projekty.

# 2 Abstrakt

Tento projekt byl proveden podle zadání z kapitoly 1, jež bylo poskytnuto vyučujícím.

Předmětem tohoto projektu je hledání **největší kliky v neorientovaném grafu**, což je jeden z typických problémů v teorii grafů **TODO: citace**. Tato problematika je krátce popsána v kapitole 1.

Pro nalezení největší kliky v neorientovaném grafu bylo vypracováno několik algoritmů, a to sice:

- Brute force algoritmus,
- Dynamický brute force algoritmus,
- Backtracking algoritmus,
- Branch and bound algoritmus.

Pro každý z těchto algoritmů byla stanovena časová komplexita nejprve teoreticky a následně také experimentálně pomocí jednoduchého programu v jazyce C.

# 3 Teoretický úvod