

# NTIN090 — Základy složitosti a vyčísitelnosti

## 6. cvičení

Petr Kučera

5. ledna 2023

Připomeňme definici problému HAMILTONOVSKÁ KRUŽNICE:

### Problém 1: HAMILTONOVSKÁ KRUŽNICE (HK)

**Instance:** Neorientovaný graf  $G = (V, E)$ .

**Otázka:** Existuje v grafu  $G$  hamiltonovská kružnice, tj. kružnice procházející všemi vrcholy?

1. Uvažme rozhodovací verzi problému OBCHODNÍHO CESTUJÍCÍHO

### Problém 2: OBCHODNÍ CESTUJÍCÍ

**Instance:** Množina měst  $C = \{c_1, \dots, c_n\}$ , hodnoty  $d(c_i, c_j) \in \mathbb{N}$  přiřazující každé dvojici měst vzdálenost a přirozené číslo  $D$ .

**Otázka:** Existuje permutace měst  $c_{\pi(1)}, c_{\pi(2)}, \dots, c_{\pi(n)}$ , pro kterou platí, že

$$\left( \sum_{i=1}^{n-1} d(c_{\pi(i)}, c_{\pi(i+1)}) \right) + d(c_{\pi(n)}, c_{\pi(1)}) \leq D?$$

a jeho optimalizační verzi

### Problém 3: OBCHODNÍ CESTUJÍCÍ

**Instance:** Množina měst  $C = \{c_1, \dots, c_n\}$ , hodnoty  $d(c_i, c_j) \in \mathbb{N}$  přiřazující každé dvojici měst vzdálenost.

**Cíl:** Nalézt permutace měst  $c_{\pi(1)}, c_{\pi(2)}, \dots, c_{\pi(n)}$ , pro kterou je

$$\left( \sum_{i=1}^{n-1} d(c_{\pi(i)}, c_{\pi(i+1)}) \right) + d(c_{\pi(n)}, c_{\pi(1)})$$

minimální.

Předpokládejme, že rozhodovací verzi problému OBCHODNÍHO CESTUJÍCÍHO umíme rozhodnout v polynomiálním čase. To znamená, že máme funkci  $\text{rozhodniOC}(C, d, D)$ , kde  $C = \{c_1, \dots, c_n\}$  je množina měst,  $d(c_i, c_j) \in \mathbb{N}$  určuje vzdálenosti mezi nimi a  $D$  je limit na vzdálenost a která rozhodne, zda existuje pořadí měst, při němž navštíví obchodní cestující všechna města, vrátí se do svého domovského města a nacestuje přitom vzdálenost nejvýš  $D$ . Popište, jak za tohoto předpokladu vyřešit v polynomiálním čase optimalizační verzi OBCHODNÍHO CESTUJÍCÍHO, tedy popište algoritmus, který nalezne pořadí měst, při němž nacestuje obchodní cestující nejkratší vzdálenost. Algoritmus bude pracovat

v polynomiálním čase za předpokladu, že volání funkce rozhodni $OC(C, d, D)$  proběhne v polynomiálním čase.

2. Rozmyslete si, jak obtížné je o dané formuli  $\varphi$  zodpovědět následující otázky v závislosti na tom, je-li  $\varphi$  KNF, DNF, či obecná výroková formule.

- (a)  $(\exists \mathbf{a})[\varphi(\mathbf{a}) = 1]$ ?
- (b)  $(\exists \mathbf{a})[\varphi(\mathbf{a}) = 0]$ ?
- (c)  $(\forall \mathbf{a})[\varphi(\mathbf{a}) = 1]$ ?
- (d)  $(\forall \mathbf{a})[\varphi(\mathbf{a}) = 0]$ ?

3. Ukažte, že pro každé  $m$  existuje formule  $\varphi$  v KNF s  $m$  klauzulemi taková, že libovolná formule  $\psi$  v DNF, která je ekvivalentní s  $\varphi$ , obsahuje alespoň  $2^m$  termů.

4. (a) Pro každou z následujících formulí popište její ekvivalentní KNF

- 1.  $v \iff \neg x$
- 2.  $v \iff x \vee y$
- 3.  $v \iff x \wedge y$
- 4.  $v \iff (x \implies y)$
- 5.  $v \iff (x \iff y)$

- (b) Uvažme formuli

$$\varphi \equiv [(x \implies y) \wedge (y \implies z)] \iff (x \implies z) \quad (1)$$

- (c) Je formule  $\varphi$  splnitelná?

- (d) Je formule  $\varphi$  tautologie?

- (e) Rozmyslete si podrobněji následující obecný postup, který k libovolné formuli  $\varphi$  zkonstruuje v polynomiálním čase formuli  $\psi$  v KNF, která je splnitelná, právě když  $\varphi$  je splnitelná.

- 1. Sestrojíme strom odvození formule  $\varphi$ .
- 2. S každým uzlem stromu asociujeme novou proměnnou.
- 3. Sémantiku dané proměnné definujeme jednou z ekvivalencí výše podle toho, jaká logická spojka je v daném uzlu.
- 4. Přepíšeme jednotlivé ekvivalence do KNF.
- 5.  $\psi$  sestrojíme jako konjunkci zkonstruovaných KNF podformulím, k nimž přidáme proměnnou kořene stromu jako jednotkovou klauzuli.

Ukažte korektnost tohoto postupu a demonstруйте jej na formuli (1). Vznikne tímto postupem formule  $\psi$ , která je tautologie?

5. Popište jednoduchý algoritmus pro 3-SAT, který v každém kroku vybere klauzuli  $(l_1 \vee l_2 \vee l_3)$  a větví na tři podpřípady dle částečných ohodnocení  $\{l_1\}$ ,  $\{\neg l_1, l_2\}$  a  $\{\neg l_1, \neg l_2, l_3\}$ . Odhadněte složitost tohoto algoritmu v závislosti na počtu proměnných  $n$  ve formuli.