

## Obecně

K řešení dvou z následujících úloh použijete algoritmus na hledání optimálních sledů. Přestože by stačilo použít jeden, použijte na každou z těchto úloh jiný algoritmus tak, aby byly vhodně vybrány k danému typu grafu, cílem je použít vždy ten nejjednodušší. Vaši volbu zdůvodněte v reportu.

### 3a. Montáž automobilu

**4 body**

**Formulace problému:** Vyrábíte speciální prototypy montovaných automobilů. Máte síť firem které vytvářejí jednotlivé části a předávají si je mezi sebou. Tyto vztahy máte zaznamenané v grafu, přičemž není pro řešení této úlohy důležité které firma předává co jaké firmě (je jedno kdo zrobené části smontuje dohromady). Vaším úkolem je zjistit časy, kdy má která firma zapojit do řetězce aby automobil byl jako celek smontovaný v nejkradším možném čase.

**Název spustitelného souboru:** car

**Vstup:** Textový soubor obsahuje vždy uživatele "Vy". Dále na každém řádku jsou uvedeni dva členové vaší sítě a doba, jak dlouho jim trvá než si předají výrobek.

```
Vy - Motor: 3
Vy - Kola: 2
Motor - Podvozek: 4
Motor - Interier: 4
Kola - Interier: 1
Kola - Skla: 2
Podvozek - Kapota: 1
Interier - Kapota: 2
Interier - Sroubky: 4
Skla - Sroubky: 2
Kapota - Sroubky: 2
```

**Výstup:** Vypište členy vaší sítě v pořadí v jakém se k nim čas kdy mají začít, tak aby v čase 0 jste měl finální výrobek.

```
Vy: 0
Kola: -2
Interier: -3
Motor: -3
Skla: -4
Kapota: -5
Podvozek: -6
Sroubky: -6
```

### 3b. Kontrola serveru

**5 body**

**Formulace problému:** V cloudu máte  $n$ -instancí vaší aplikace. Máte dávku dat kterou musíte postupně zpracovat na každém instanci vaší aplikace. Data se v tomto výpočtu musí předávat mezi virtuálními uzly, nelze úlohu spustit paralelně a vzhledem k objemu dat hraje důležitou roli čas jejich přenosu z jednoho počítače na jiný. Váš cloud je hostovaný po celém světě a vy znáte čas který zabere přenos mezi každými dvě-mi instancemi. Tuto informaci máte zadanou na vstupu. Najděte takovou posloupnost severů aby čas přenosu byl minimální.

**Název spustitelného souboru:** server

**Vstup:** Na každém řádku je uvedena jedna cesta mezi dvěma servery, číslo udává čas, který je potřebný na přenos dat.

A - B: 1  
B - C: 2  
C - D: 1  
D - A: 3  
A - C: 3  
B - D: 2

**Výstup:** Výstupem algoritmu bude posloupnost virtuálních serverů a celkový čas, který zabere přenos dat:

A - B - C - D: 4

### 3c. Spartan race

**5 body**

**Formulace problému:** V jiném závodě máte za úkol doběhnout ze startu do cíle, nicméně jak název napovídá je tento závod postaven na výdrži. Vaším úkolem je tedy najít takovou cestu, která bude energeticky nejvýhodnější. Závodní trať se skládá z křižovatek vzájemně propojených cestami a vy si můžete zvolit, kterou cestou se vydáte. Cesty jsou hodnoceny podle náročnosti:

Běh po rovině = ohodnocení 0

Běh z kopce = ohodnocení kladné, čím delší tím kladnější

Běh do kopce = ohodnocení záporné, čím delší tím zápornější

Na některých křižovatkách je umístěno občerstvení (+1), které vám pomůže trasu zdolat. Závod je natolik drsný, že je energeticky nejvýhodnější dostat se do cíle co nejdříve, bez opakovaného navštěvování křižovatek.

**Název spustitelného souboru:** race

**Vstup:** Textový soubor obsahuje na každém řádku jednu křižovátku (pokud je označená +, znamená to, že je tam bonus), a seznam křižatek, do kterých se lze z této křižovátky dostat, v závorce je uvedena náročnost. První křižovátka je startovní, poslední křižovátka je cílová.

A: B(2), C(-3)

B+: C(3), D(-2)

C: B(-4), D(-3)

D: A(-4)

**Výstup:** Výstupem algoritmu bude seznam projetych křižovatek v pořadí, v jakém mají být proběhnuty a cena cesty:

A - B - C - D: 3