# Ćwiczenia 11: Maksymalny przepływ

**[dla osób z najdalej 2 plusami] Zadanie 1. (fabryki i sklepy)** Mamy dany graf skierowany $G = (V, E)$ oraz funkcję $c\colon E \to \mathbb{N}$ opisującą przepustowość każdej krawędzi (liczbę jednostek towaru na godzinę, które mogą się przemieszczać krawędzią). Poza tym mamy dany zbiór wierzchołków-fabryk $S = \{s_1, \dots, s_n\}$ oraz zbiór wierzchołków-sklepów $T = \{t_1, \dots, t_m\}$. Dla każdej fabryki $s_i$ znamy liczbę $p_i$ określającą ile jednostek towaru na godzinę fabryka może maksymalnie produkować. Jednocześnie dla każdego sklepu $t_j$ mamy liczbę $q_j$, która mówi ile jednostek towaru na godzinę musi do tego sklepu docierać. Proszę podać algorytm, który sprawdza, czy da się zapewnić, żeby do każdego sklepu docierało z fabryk dokładnie tyle jednostek towaru ile sklep wymaga jednocześnie nie zmuszając żadnej fabryki do przekroczenia swoich możliwości produkcyjnych i nie przekraczając przepustowości żadnej z krawędzi.

**[dla osób z najdalej 2 plusami] Zadanie 2. (maksymalny przepływ w grafie nieskierowanym)** Proszę wskazać algorytm, który znajduje maksymalny przepływ między źródłem i ujściem w grafie nieskierowanym. Proszę użyć algorytmu z wykładu—dla grafów skierowanych, gdzie między każdą parą wierzchołków jest najwyżej jedna krawędź—jako czarnej skrzynki.

**Zadanie 3. (spójność krawędziowa)** Dany jest graf nieskierowany $G = (V, E)$. Mówimy, że spójność krawędziowa $G$ wynosi $k$ jeśli usunięcie pewnych $k$ krawędzi powoduje, że $G$ jest niespójny, ale usunięcie dowolnych $k - 1$ krawędzi nie rozspójnia go. Proszę podać algorytm, który oblicza spójność krawędziową danego grafu.

**Zadanie 4. (Formuły logiczne z dwoma wystąpieniami zmiennej)** Dana jest formuła logiczna postaci: $C_1 \wedge C_2 \wedge \cdots \wedge C_m$, gdzie każda $C_i$ to klauzula będąca alternatywą zmiennych i/lub ich zaprzeczeń. Wiadomo, że każda zmienna występuje w formule dokładnie dwa razy, raz zanegowana i raz niezanegowana. Na przykład poniższa formuła stanowi dopuszczalne wejście:

$$(x \vee y \vee z) \wedge (\overline{y} \vee w) \wedge (\overline{z} \vee v) \wedge (\overline{x} \vee \overline{w}) \wedge (\overline{v}).$$

Proszę podać algorytm, który oblicza takie wartości zmiennych, że formuła jest prawdziwa.

**Zadanie 5. (skojarzenie na drzewie)** Proszę podać algorytm, który mając na wejściu drzewo oblicza skojarzenie o maksymalnej liczności.

**Zadanie 6. (odległość Wassersteina/Earthmover's distance)** Dane są dwa wektory $x = (x_1, \dots, x_n) \in \mathbb{N}^n$ oraz $y = (y_1, \dots, y_n) \in \mathbb{N}^n$ takie, że $\sum_{i=1}^n x_i = \sum_{i=1}^n y_i = T$. Pojedyncza operacja *przenoszenia* powoduje zmniejszenie jednego pola wektora i zwiększenie pola obok (ale żadne pole nigdy nie może zmniejszyć się poniżej zera). Odległość Wassersteina tych wektorów—zapisywana jako $d(x, y)$—zdefiniowana jest jako najmniejsza liczba operacji przenoszenia potrzebnych do zamiany wektora $x$ w wektor $y$. (Intuicyjnie wektory opisują ile kamyczków jest w poszczególnych polach wektora i pojedyncza operacja przenosi jeden kamyczek o jedno pole; interesuje nas minimalna liczba ruchów). Proszę zaproponować możliwie jak naszybszy algorytm obliczający odległość Wassersteina dwóch wektorów.

**Zadanie 7. (odległość Wassersteina na macierzach)** Proszę zaproponować algorytm obliczający odległość Wassersteina na macierzach o wymiarze $n \times n$. (Załóżmy, że jest zdefiniowana tak, jak na wektorach i każdy kamyczek można przesunąć w górę, dół, na lewo, lub na prawo; czasem używa się trochę innych definicji)

**Zadanie 8. (rozłączne ścieżki)** Dany jest graf skierowany $G = (V, E)$ oraz wierzchołki $s$ i $t$. Proszę zaproponować algorytm znajdujący maksymalną liczbę rozłącznych (wierzchołkowo) ścieżek między $s$ i $t$.