Ćwiczenia 8: Algorytmy dynamiczne, zachłanne i podstawy grafowych

[osoby z $\leqslant 1$ plusem] Zadanie 1. (kolorowanie grafu) Mamy dany nieskierowany graf $G = (V, E)$. Dla każdego wierzchołka v należy wybrać liczbę naturalną $f(v)$ (nazywaną kolorem v) tak, żeby dla każdej krawędzi $\{x, y\} \in E$ zachodzilo, ze $f(x) \neq f(y)$. Należy użyć jak najmniej kolorów (czyli zbior $f(V)$ powinien mieć minimalną liczność). Problem jest NP-zupełny więc nie istnieje optymalny algorytm wielomianowy (o ile P jest różne od NP). Proszę podać algorytm zachłanny, który używa najwyżej $D + 1$ kolorów, gdzie D to maksymalny stopień wierzchołka w G.

[osoby z $\leqslant 1$ plusem] Zadanie 2. (BFS i najkrótsze ścieżki) Proszę zaimplementować algorytm BFS tak, żeby znajdował najkrótsze ścieżki w grafie i następnie, żeby dało się wypisać najkrotszą ścieżkę z zadanego punktu startowego do wskazanego wierzchołka.

Zadanie 3. (ładowanie promu) Dana jest tablica $A[n]$ z długościami samochodów, które stoją w kolejce, żeby wjechać na prom. Prom ma dwa pasy (lewy i prawy), oba długości D. Proszę napisać program, który wyznacza, które samochody powinny pojechać na który pas, żeby na promie zmieściło się jak najwięcej aut. Auta muszą wjeżdżac w takiej kolejności, w jakiej są podane w tablicy A.

Zadanie 4 (podprzedziały) Dany jest zbiór przedziałów $I = \{[a_1, b_1], \ldots, [a_n, b_n]\}$. Mówimy, że przedział $[a_i, b_i]$ deaktywuje przedział $[a_j, b_j]$ jeśli $a_i \geqslant a_j$ oraz $b_i \leqslant b_j$ (czyli pierwszy jest podzbiorem drugiego). Proszę zaproponować algorytm, który znajduje podzbiór I o maksymalnym rozmiarze taki, że żaden przedział nie jest deaktywowany przez inny (innymi słowy, należy dany przedział usunąć jeśli jest nadzbiorem innego, a spośród identycznych przedziałów usunąć wszystkie poza jednym).

Zadanie 5 (najkrótszy ciąg zadań) Dany jest zbiór zadań $T = \{t_1, \ldots, t_n\}$. Każde zadanie t_i ma podany czas rozpoczęcia $s_i \in \mathbb{R}$ oraz czas zakończenia $e_i \in \mathbb{R}$. Proszę zaproponować algorytm (bez implementacji), który znajduje taki podzbiór k zadań (gdzie $k \in \mathbb{N}$ to dany parametr wejściowy), że (a) żadne dwa zadania na siebie nie nachodzą oraz (b) czas jaki mija od rozpoczęcia najwcześniejszego zadania do zakończenia najpózniejszego jest minimalny. Jeśli podzbioru rozmiaru k spełniajacego warunki zadania nie ma, to algorytm powinien to stwierdzić. Algorytm powinien być jak najszybszy. Można założyć, że żaden przedział nie jest podzbiorem innego.

Zadanie 6. (krawędzie 0/1) Dana jest mapa kraju w postaci grafu $G = (V, E)$. Kierowca chce przejechać z miasta (wierzchołka) s to miasta t. Niestety niektóre drogi (krawędzie) są płatne. Każda droga ma taką samą jednostkową opłatę. Proszę podać algorytm, który znajduje trasę wymagającą jak najmniejszej liczby opłat. W ogólności graf G jest skierowany, ale można najpierw wskazać algorytm dla grafu nieskierowanego.

Zadanie 7. (bezpieczny przelot) Dany jest graf $G = (V, E)$, którego wierzchołki reprezentują punkty nawigacyjne nad Bajtocją, a krawędzie reprezentują korytarze powietrzne między tymi punktami. Każdy korytarz powietrzny $e_i \in E$ powiązany jest z optymalnym pułapem przelotu $p_i \in \mathbb{N}$ (wyrażonym w metrach). Przepisy dopuszczają przelot danym korytarzem jeśli pułap samolotu różni się od optymalnego najwyżej o t metrów. Proszę zaproponować algorytm (bez implementacji), który sprawdza czy istnieje możliwość przelotu z zadanego punktu $x \in V$ do zadanego punktu $y \in V$ w taki sposób, żeby samolot nigdy nie zmieniał pułapu. Algorytm powinien być poprawny i możliwie jak najszybszy. Proszę oszacować jego złożoność czasową.

Zadanie 8 (Horn-SAT) Na wejściu dana jest formuła logiczna F w postaci CNF (conjunctive normal form), czyli $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, gdzie każde C_i to tak zwana klauzula, czyli alternatywa zmiennych logicznych lub ich negacji. Formuła w postaci CNF jest Hornowska jeśli każda klauzula zawiera najwyżej jedną niezanegowaną zmienną. Przykłady Hornowskich formuł CNF to: $(x) \wedge (\overline{x} \vee y) \wedge (\overline{x} \vee \overline{y} \vee z)$, $(\overline{x} \vee y) \wedge (\overline{x} \vee y)$, czy $(x) \wedge (\overline{x})$. Proszę zaproponować wielomianowy algorytm, który sprawdza czy Hornowska formuła w postaci CNF jest spełnialna (to znaczy, czy da sie przypisac wartości logiczne zmiennym tak, żeby formuła miala wartość *prawda*).

Zadanie 9. (bitoniczny problem komiwojażera) Mamy dane punkty na płaszczyźnie. Komiwojażer może się poruszać najpierw tylko w prawo a potem tylko w lewo (oraz dowolnie góra/dół). Należy podać algorytm dynamiczny znajdujący optymalną trasę komiwojażera, w której odwiedza wszystkie zadane punkty i wraca do punktu startu (czyli do punktu maksymalnie na lewo).

Zadanie 10. (problem komiwojażera) Na wejściu dostajemy zbiór n miast oraz macierz odległości między każdą parą miast (powiedzmy, że d_{ij} to odległość z miasta i do j, oraz $d_{ij} = d_{ji}$). Zadanie polega na obliczeniu najkrótszej trasy, która zaczyna się w mieście 1, kończy w mieście 1 i przebiega przez wszystkie pozostałe, żadnego nie powtarzając.