Lista zadań. Nr 2. 9 marca 2023

Algorytmy i Struktury Danych

IIUWr.

1. (0pkt) Przeczytaj notatkę do wykładu o algorytmach zachłannych.
2. (1pkt) Udowodnij, że algorytm Kruskala znajduje minimalne drzewa spinające poprzez przyrównanie tych drzew do drzew optymalnych.
3. (1pkt) Danych jest n odcinków $I_j = \langle p_j, k_j \rangle$, leżących na osi OX, $j = 1, \ldots, n$. Ułóż algorytm znajdujący zbiór $S \subseteq \{I_1, \ldots, I_n\}$, nieprzecinających się odcinków, o największej mocy.
4. (1pkt) Rozważ następującą wersję problemu wydawania reszty: dla danych liczb naturalnych a, b ($a \leq b$) chcemy przedstawić ułamek $\frac{a}{b}$ jako sumę różnych ułamków o licznikach równych 1. Udowodnij, że algorytm zachłanny zawsze daje rozwiązanie. Czy zawsze jest to rozwiązanie optymalne (tj. o najmniejszej liczbie składników)?
5. (1,5pkt) Ułóż algorytm, który dla danego n-wierzchołkowego drzewa i liczby k, pokoloruje jak najwięcej wierzchołków tak, by na każdej ścieżce prostej było nie więcej niż k pokolorowanych wierzchołków.
6. (2pkt) Ułóż algorytm, który dla danego spójnego grafu G oraz krawędzi e sprawdza w czasie $O(n + m)$, czy krawędź e należy do jakiegoś minimalnego drzewa spinającego grafu G. Możesz założyć, że wszystkie wagi krawędzi są różne.
7. (2pkt) System złożony z dwóch maszyn A i B wykonuje n zadań. Każde z zadań wykonywane jest na obydwu maszynach, przy czym wykonanie zadania na maszynie B można rozpocząć dopiero po zakończeniu wykonywania go na maszynie A. Dla każdego zadania określone są dwie liczby naturalne a_i i b_i określające czas wykonania i-tego zadania na maszynie A oraz B (odpowiednio). Ułóż algorytm ustawiający zadania w kolejności minimalizującej czas zakończenia wykonania ostatniego zadania praz maszynę B.
8. (2pkt) Niech $T = (V, E)$ będzie drzewem a $P(u, v)$ niech oznacza ścieżkę w T (rozumianą jako zbiór krawędzi) łączącą wierzchołki u i v.

   Ułóż algorytm, który dla drzewa T znajduje trzy wierzchołki a, b, c, dla których zbiór $\{e \in E : e \in P(a, b) \cup P(a, c) \cup P(b, c)\}$ jest maksymalnie duży.
9. (2pkt) Operacja $swap(i, j)$ na permutacji powoduje przestawienie elementów znajdujących się na pozycjach i oraz j. Koszt takiej operacji określamy na $|i - j|$. Kosztem ciągu operacji *swap* jest suma kosztów poszczególnych operacji.

   Ułóż algorytm, który dla danych π oraz σ - permutacji liczb $\{1, 2, \ldots, n\}$, znajdzie ciąg operacji *swap* o najmniejszym koszcie, który przekształca permutację π w permutację σ.
10. (1pkt) Na wykładzie przedstawiono zachłanny algorytm dla problemu *Pokrycia zbioru*, znajdujący rozwiązania, które są co najwyżej $\log n$ razy gorsze od rozwiązania optymalnego.

    Pokaż, że istnieją dane, dla których rozwiązania znajdowane przez ten algorytm są blisko $\log n$ gorsze od rozwiązań optymalnych.
11. (**Z** 2pkt) *Wariancją* wag krawędzi grafu $G = (V, E; w)$ nazywamy wielkość

    $$s(G) = \frac{1}{|E|} \sum_{e \in E} (w(e) - \overline{w})^2$$

    gdzie $\overline{w}$ jest średnią wag krawędzi (a więc $\overline{w}$ jest równe $\frac{1}{|E|} \sum_{e \in E} w(e)$).

Ułóż algorytm, który dla zadanego grafu znajduje drzewo spinające T o minimalnej wartości $s(T)$.

UWAGA: Nawet rozwiązania o dużej złożoności (zwykle nie akceptowalnej na AiSD) mogą okazać się interesujące. Uprzedzając Wasze pytania: rozwiązania o złożoności wykładniczej nie będą interesujące:-(

Zadania dodatkowe - do samodzielnego rozwiązywania lub na repetytorium

1. (2pkt) Dla ważonego drzewa $T = (V, E; c)$, gdzie $c : V \to \mathcal{R}_+$, określamy jego *zewnętrzną długość* $EL(T)$ jako:
$$EL(T) = \sum_{v-li\acute{s}\acute{c}\in T} c(v) \cdot d(v),$$
gdzie $d(v)$ jest długością ścieżki od korzenia do liścia v (mierzoną liczbą krawędzi na ścieżce).

   Rozważmy następujący problem. Dany jest n-elementowy zbiór $\{w_1, \ldots, w_n\}$ dodatnich liczb rzeczywistych. Zadaniem jest znalezienie ważonego drzewa binarnego T o n liściach, takiego, że każda liczba w_i jest wagą dokładnie jednego liścia oraz T ma minimalną wagę $EL(T)$ pośród wszystkich drzew o tej własności.

2. (2pkt) Ułóż algorytm, który dla danych liczb naturalnych a i b, sprawdza, czy zachłanna strategia dla problemu wydawania reszty jest poprawna, gdy zbiór nominałów jest równy $X = \{1, a, b\}$.

3. (1pkt) Wykaż, że zachłanny algorytm wydawania reszty, w sytuacji gdy monety mają nominały $c^0, c^1, \ldots, c^k$ dla pewnych stałych naturalnych $c > 1$ i $k \geq 1$, daje optymalne rozwiązanie.

4. (1pkt) Udowodnij, że spójny graf, którego wszystkie krawędzie mają różne wagi, posiada dokładnie jedno minimalne drzewo spinające.

5. (1pkt) Udowodnij poprawność algorytmu Prima.

6. (1,5pkt) Udowodnij poprawność algorytmu Boruvki (Sollina).

7. (1pkt) Ułóż algorytm rozwiązujący następujący problem szeregowania zadań dla c procesorów:
   Dane: t_i - czas obsługi i-tego zadania ($i = 1, 2, ..., n$).
   Problem: każde z zadań przypisać do jednego z c procesorów oraz ustalić kolejność wykonywania zadań przez każdy z procesorów tak, by zminimalizować wartość:
   $T = \sum_{i=1}^{n}$ (czas przebywania i-tego zadania w systemie).

   Udowodnij, że Twój algorytm zawsze znajduje optymalne rozwiązanie.

   Jaki jest czas działania Twojego algorytmu?

8. (2pkt) Rozważamy grafy skierowane, w których każda para wierzchołków połączona jest przynajmniej jedną krawędzią. Podaj algorytm wyznaczający dla takiego grafu *ścieżkę Hamiltona*, tj. ścieżkę przechodzącą dokładnie jeden raz przez wszystkie wierzchołki. Udowodnij, że Twój algorytm działa poprawnie.

9. (2pkt) Udowodnij, że w grafie nieskierowanym o n wierzchołkach, w którym każdy wierzchołek ma co najmniej $n/2$ sąsiadów, istnieje ścieżka Hamiltona. Podaj algorytm, który dla takich grafów znajduje tę ścieżkę.

10. (1pkt) Przypomnij sobie algorytm Dijkstry znajdowania najkrótszych ścieżek od zadanego wierzchołka do wszystkich pozostałych wierzchołków. Udowodnij jego poprawność.