

## ALGORYTMY I STRUKTURY DANYCH

IHUWr.

0. (0pkt) Przeczytaj notatkę do wykładu o algorytmach zachłannych.
1. (1pkt) Przeprowadź dowód poprawności algorytmu Kruskala, który przyrównuje ciąg krawędzi drzewa wybranych przez algorytm Kruskala z ciągiem krawędzi minimalnego drzewa spinającego (otrzymanego przez jakiś algorytm optymalny). W dowodzie nie powołuj się na własności typu *cut property* czy *cycle property*.
2. (1pkt) Danych jest  $n$  odcinków  $I_j = \langle p_j, k_j \rangle$ , leżących na osi OX,  $j = 1, \dots, n$ . Ułóż algorytm znajdujący zbiór  $S \subseteq \{I_1, \dots, I_n\}$ , nieprzecinających się odcinków, o największej mocy.
3. (1,5pkt) Rozważ wersję problemu wydawania reszty, w którym  $i$ -ty nominal ma wartość równą  $i$ -tej liczbie Fibonacciego ( $i = 1, 2, \dots$ ). Ułóż algorytm, który, wydając resztę, używa co najwyżej jednej monety każdego nominalu. Czy taki algorytm jest w stanie wydać każdą kwotę? Czy liczba monet wydana przez Twój algorytm jest zawsze optymalna (tj. minimalna)?
4. (2pkt) Masz początkowo do dyspozycji  $m$  monet o nominale 1 i nieskończoną liczbę monet o nominale 100. W kolejnych  $n$  dniach masz zrobić zakupy w ulubionym sklepie (w  $i$ -tym dniu zakup o wartości  $c_i$ ). Jeśli w  $i$ -tym dniu nie odliczysz dokładnej kwoty (tj.  $c_i$ ), kasa wyda Ci dokładną wartość reszty, używając minimalnej ilości monet (kasa także używa jedynie monet o nominalach 1 i 100), a Twoja "atrakcyjność klienta" zostanie zmniejszona o wartość  $x \cdot w_i$ , gdzie  $x$  jest liczbą monet wydanych przez kasę a  $w_i$  jest współczynnikiem używanych przez sklep w  $i$ -tym dniu.
- Ułóż algorytm obliczający o ile co najmniej zmniejszy się Twoja atrakcyjność klienta po dokonaniu wszystkich zakupów. Możesz przyjąć, że  $c_i$  oraz  $w_i$  są liczbami naturalnymi nie większymi od  $n$ .
5. (1,5pkt) Ułóż algorytm, który dla danego grafu  $G = (V, E)$  oraz liczby naturalnej  $k$  znajdzie możliwie największy podzbiór  $V' \subseteq V$ , taki, że dla każdego wierzchołka  $v \in V'$  zachodzi:
- $$|\{u \in V' : \{v, u\} \in E\}| \geq k \text{ oraz } |\{u \in V' : \{v, u\} \notin E\}| \geq k.$$
6. (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.
7. (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.
8. (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  $\pi$  oraz  $\sigma$  - permutacji liczb  $\{1, 2, \dots, n\}$ , znajdzie ciąg operacji  $swap$  o najmniejszym koszcie, który przekształca permutację  $\pi$  w permutację  $\sigma$ .
9. (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.