

# Algorytmy online

## Lista 12

**Zadanie 1 (2 pkt.)** W problemie drzewa Steinera dany jest  $n$ -wierzchołkowy graf z wyróżnionym wierzchołkiem  $r$ . Wejście jest ciągiem wierzchołków grafu; jeśli na wejściu pojawia się wierzchołek  $v$ , algorytm musi zwrócić pełną ścieżkę łączącą  $r$  z  $v$ . W każdym momencie rozwiązaniem algorytmu jest zatem pewne drzewo łączące  $r$  i wszystkie żądania, zaś jego kosztem jest sumaryczna waga krawędzi tego drzewa. Skonstruuj randomizowany  $O(\log n)$ -konkurencyjny algorytm dla tego problemu.

*Wskazówka:* Skonstruuj 1-konkurencyjny algorytm online ALG, gdy grafu będącego drzewem. Przybliż graf losowym drzewem konstrukcją FRT z wykładu i uruchom na tym drzewie ALG.

**Zadanie 2 (2 pkt.)** Mamy do dyspozycji  $k$ -konkurencyjny algorytm dla problemu  $k$  serwisantów dla drzew.<sup>1</sup> Dany jest  $n$ -wierzchołkowy graf. Skonstruuj randomizowany  $O(k \cdot \log n)$ -konkurencyjny algorytm dla tego grafu.

**Zadanie 3 (2 pkt.)** Drzewo  $T$  otrzymane w przedstawionej na wykładzie konstrukcji FRT składa się z liści odpowiadających wierzchołkom grafu  $V$  i wierzchołków wewnętrznych  $V'$ . Pokaż, jak na podstawie  $T$  skonstruować drzewo  $T'$ , w którym zbiór wierzchołków jest równy  $V$  a odległości w  $T'$  między dowolną parą  $v, w \in V$  różnią się od odległości w  $T$  co najwyżej  $O(1)$  razy.

*Wskazówka:* usuwaj niektóre krawędzie.

**Zadanie 4** W pewnym kraju linia brzegowa jest prostą oddzielającą brzeg od morza. Pewien kapitan odpłynął od brzegu na odległość 1 km, rzucił kotwicę i zrobił wielką balangę. Kiedy następnego dnia obudził się na kacu, nie wiedział skąd ani jak się tu wziął; pamiętał tylko, że od odpłynął od brzegu na odległość 1 km. Oczywiście panuje gęsta mgła a kapitan jest głuchy i ślepy. Na pokładzie nie ma krowy, która mogłaby pomóc znaleźć mu brzeg, jest to zatem zadanie dla Ciebie. (Jest to wariant poszukiwania prostej na płaszczyźnie: tym razem wiadomo tylko, że prosta jest w odległości równej 1).

- (2 pkt.)** Skonstruuj deterministyczny algorytm, którego współczynnik ścisłej konkurencyjności wynosi dokładnie  $1 + 2 \cdot \pi$ .
- (2 pkt.)** Skonstruuj deterministyczny algorytm, którego współczynnik ścisłej konkurencyjności jest mniejszy niż  $1 + 2 \cdot \pi$ .

Marcin Bieńkowski

---

<sup>1</sup>Przedstawiony na wykładzie algorytm DOUBLE COVERAGE można łatwo rozszerzyć do drzew: do żądania przesuwa on ze stałą prędkością “niezasłoniętych” serwisantów. (Serwisant  $x$  jest niezasłonięty jeśli ścieżka od żądania do  $x$  nie zawiera innych serwisantów).