

Lab 03 (07.03.2019)

Część 1

Sprawdzenie czy podany ciąg stopni jest grafowy

[0.5 pkt]

Ciąg jest grafowy, jeżeli istnieje graf o zadanym ciągu stopni wierzchołków.

Do sprawdzenia czy ciąg (d_1, d_2, \dots, d_n) jest grafowy można użyć następującego twierdzenia *Havel'a-Hakimi'ego*:

Nierosnący ciąg (d_1, d_2, \dots, d_n) jest grafowy wtedy i tylko wtedy gdy ciąg

$(d_2 - 1, d_3 - 1, \dots, d_{d_1+1} - 1, d_{d_1+2}, d_{d_1+3}, \dots, d_n)$ również jest grafowy.

Uwaga:

- ciąg składający się z samych 0 jest grafowy

Część 2

Konstruowanie grafu na podstawie podanego ciągu grafowego

[1.5 pkt]

Zaimplementuj algorytm zachłanny na podstawie podanego twierdzenia Havel'a-Hakimi'ego.

Część 3

Wyznaczanie minimalnego drzewa (bądź lasu) rozpinającego algorytmem Kruskala

[2 pkt]

Schemat algorytmu Kruskala

1. wrzucić wszystkie krawędzie do "wspólnego worka"
2. wyciągać z "worka" krawędzie w kolejności wzrastających wag
 - o jeśli krawędź można dodać do drzewa to dodawać, jeśli nie można to ignorować
 - o punkt 2 powtarzać aż do skonstruowania drzewa (lasu) lub wyczerpania krawędzi

Parametry:

- *graph* - graf wejściowy
- *min_weight* - waga skonstruowanego drzewa (lasu)

Wynik:

skonstruowane minimalne drzewo rozpinające (albo las)

Uwagi:

1. Metoda uruchomiona dla grafu skierowanego powinna zgłaszać wyjątek *ArgumentException*
2. Graf wejściowy pozostaje niezmieniony
3. Wykorzystać klasę *UnionFind* z biblioteki *Graph*
4. Wykorzystać klasę *EdgesMinPriorityQueue* z biblioteki *Graph*
5. Jeśli graf *graph* jest niespójny to metoda wyznacza las rozpinający (składający się z drzew rozpinających kolejnych składowych spójności)