

Praca domowa 03 – highway

Michał Szczygieł

Wstęp

Zadanie składa się z 2 plików:

- Main.java
- Highway.java

Klasa Main

Klasa ta odpowiedzialna jest za wczytanie danych z pliku, oraz przekazanie poprawnie odebranych danych do klasy Highway.

Metody:

`private static ArrayList<Integer> readFile(File file)` - Czytanie danych z pliku wejściowego oraz zwrócenie wyniku w postaci listy stringów.

Klasa Highway

Klasa Highway zawiera między innymi dwa klasy wewnętrzne, w celu ułatwienia dostępu do danych:

```
public class Graph
public class Traversal
```

Klasa Graph odzwierciedla reprezentację autostrad jako rzeczywistego grafu. Zawiera także metody wyszukiujące, dodające usuwające wierzchołki (miasta) z grafu (z wyspy).

Klasa Traversal jest pewnym zbiorem metod operujących na grafie. Metody jakie to wyszukiwanie cykli, ścieżek w grafie, dzielenia na mniejsze cykle.

Realizacja algorytmu możliwości budowy autostrad

Sama klasa Highway wykorzystująca obiekty powyższych klas rozwiązuje problem poprzez sprawdzanie planarności grafu. Każdemu miastu przyporządkowuje wierzchołek w grafie, a autostrada stanowi krawędź pomiędzy dwoma wierzchołkami. Dane wpisuje do listy sąsiedztwa. Następnie sprawdzam, podstawowe założenia odnoszące się do planarności, takie jak $|E| \leq 3 \cdot |V| - 6$. Gdzie E to ilość wierzchołków, a V krawędzi.

Następnie sprawdzam i dzielę dany graf na kolejne pod grafy. W kolejnym etapie „koloruje” graf. Jeśli program sprawdzi że graf nie jest planarny ustawia status, o możliwości budowy autostrady.

W moim wypadku złożoność algorytmu osiąga $O(n^3)$.

Dyskusja

Można było by zastosować „lżejsze rozwiązanie” poprzez zastosowanie algorytmu drzew zrównoważonych (AVL lub czerwono-czarnych) o złożoności $O(\log k)$. Dlaczego wybrałem moje rozwiązanie, istnieje powód prosty, brakło mi czasu ponieważ brakło mi czasu na zaimplementowanie powyższego rozwiązania.