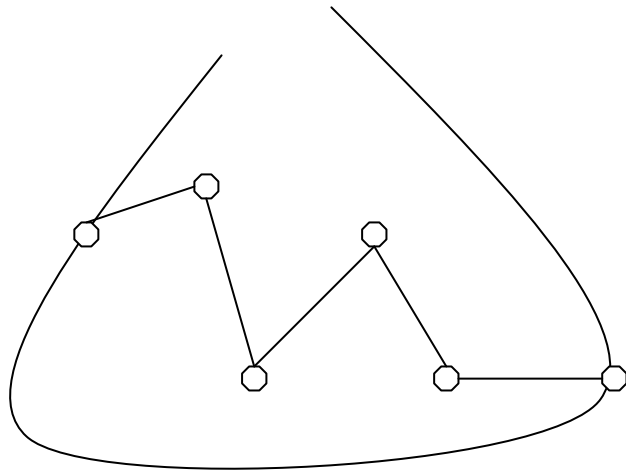


Praca domowa 03 – highway

Termin zwrotu : 3 listopada godz 23.00

Zadanie uznaje się za zaliczone, gdy praca oceniona zostanie na co najmniej 6 pkt.

Dwa miasta leżące odpowiednio na zachodnim oraz wschodnim brzegu półwyspu połączone zostały drogą przechodzącą przez wszystkie n miast leżących na półwyspie w taki sposób, iż droga przechodzi przez każde z nich, wyznaczając równocześnie ich numerację (czyli miasto nr 1 na wybrzeżu zachodnim, miasto oznaczone jako n na wschodnim).



Planuje się budowę k autostrad łączących z góry wskazane pary miast ze sobą. Nowo budowane autostrady nie mogą się wzajemnie przecinać, ani też przecinać istniejącej już drogi. Mogą być zatem prowadzone zależnie od potrzeb północną lub południową stroną istniejącej drogi.

Z pliku wejściowego (wskazywanego przez parametr `<input_file>` programu) należy wczytać kolejno ilość miast ($n > 2$), ilość planowanych do budowy autostrad ($k > 1$) oraz k kolejnych par liczb i, j gdzie i, j oznaczają numery miast, które winny zostać połączone autostradą (a więc i oraz j przyjmują wartości z przedziału $1, 2 \dots n$). Długość pojedynczej linii pliku wejściowego ani ilość liczb w linii nie jest w jakikolwiek sposób określona (format swobodny).

Wynikiem działania programu jest wartość 1 – jeżeli planowana sieć połączeń jest możliwa do realizacji, 0 w przeciwnym przypadku.

Program ma być zapisany wyłącznie w dwóch plikach `Highway.java` – zawierającym implementację algorytmu, oraz `Main.java` – zawierającym program główny. Program nie może korzystać z jakichkolwiek bibliotek zewnętrznych.

Proces kompilacji musi być możliwy z użyciem komendy

```
javac -Xlint Main.java Highway.java
```

Uruchomienie programu winno być możliwe z użyciem komendy

```
java Main <input_file>
```

Przykładowy wynik końcowy :

Wynik : 1

Wymagania :

- Klasa implementująca elementy programu głównego winna zostać zdefiniowana w pliku `Main.java`
- Klasa implementująca mechanizm rozkładu winna być zdefiniowana w pliku `Highway.java`
- W pliku `README.pdf` winien być zawarty opis organizacji struktur danych, szczegółowy opis algorytmu oraz analiza/dyskusja złożoności obliczeniowej zaproponowanego rozwiązania.

Sposób oceny :

- 1 pkt – **Kompilacja** : każdy z plików winien być kompilowany bez jakichkolwiek błędów lub ostrzeżeń (w sposób omówiony wyżej)
- 1 pkt – **Wykonanie** : program powinien wykonywać się bez jakichkolwiek błędów i ostrzeżeń (dla pliku danych wejściowych zgodnych z wyżej zamieszczoną specyfikacją) z wykorzystaniem omówionych wyżej parametrów linii komend
- 2 pkt – **README** : plik `README.pdf` dokumentuje w sposób kompletny i właściwy zarówno struktury danych, strategię poszukiwania rozwiązania (algorytm) oraz szacowaną złożoność obliczeniową przyjętego rozwiązania.
- 1 pkt – **Komentarze wewnętrzne** : czy program jest skomentowany w sposób zapewniający zrozumienie jego działania, oraz wyjaśniający warunki, które muszą zachodzić przed i po wykonaniu każdej z funkcji.
- 1 pkt – **Styl kodowania** : Styl oceniany jest w oparciu o wskaźniki NCSS oraz CCN (patrz dokument *‘Zasady oznaczania i przesyłania prac’*) oraz dodatkowo z uwzględnieniem następujących elementów : Czy funkcje i zmienne posiadają samo-wyjaśniające nazwy ? Czy podział na funkcje ułatwia czytelność i zrozumiałość kodu ? Czy funkcje eliminują (redukują) powtarzające się bloki kodu ? Czy wcięcia, odstępy, wykorzystanie nawiasów itp. (formatowanie kodu) są spójne i sensowne ?
- 4 pkt – **Poprawność algorytmu** : czy rozwiązanie problemu zostało zaprojektowane i zaimplementowane poprawnie, przy czym za merytorycznie poprawny algorytm można uzyskać dwa punkty, a dwa kolejne za te wyróżniające elementy/cechy rozwiązania (trafność doboru struktury danych, efektywność algorytmu itp.).