1 Treść zadania

Napisać program, do sprawdzania, czy graf nieskierowany jest dwudzielny. Plik z grafem ma następującą postać:

- każda krawędź jest podana w osobnej linii; podane są dwa wierzchołki, które łączy krawędź.
- W pliku mogą wystąpić puste linie.
- W linii mogą wystąpić dodatkowe (nadmiarowe) znaki białe.

Program wypisuje do pliku wyjściowego zadany graf i komunikat, czy jest to graf dwudzielny, czy nie. Jeżeli zadany graf jest dwudzielny, program wypisuje wierzchołki obu grup grafu.Program uruchamiany jest z linii poleceń z potrzebnymi przełącznikami, natomiast uruchomienie programu bez parametrów powoduje wypisanie krótkiej instrukcji.

2 Analiza zadania

Zagadnienie przedstawia problem sprawdzenia dwudzielności grafu zapisanego w pliku.

2.1 Struktury danych

W programie wykorzystano strukturę do przechowywania wartości. Struktura umożliwia przechowywanie różnych danych, jest to wykorzystane w programie, ponieważ każda struktura wierzchołka posiada informację o kolorze wierzchołka, jego identyfikator oraz informację z jakimi wierzchołkami dany wierzchołek jest połączony. Umożliwia to sprawne przemieszczanie się z jednego wierzchołka na kolejny.

2.2 Algorytmy

Program sprawdza czy graf jest dwudzielny przy wykorzystaniu algorytmu przeszukiwania wszerz (BFS). Przechodzenie grafu tym algorytmem rozpoczyna się od zadanego wierzchołka i polega na odwiedzeniu wszystkich osiągalnych z niego wierzchołków. Przechodzenie wszerz umożliwia analizę kolorów kolejnych wierzchołków i stwierdzenie czy graf jest dwudzielny.

3 Specyfikacja zewnętrzna

Program jest uruchamiany z linii poleceń. Należy przekazać do programu nazwy plików tekstowych: wejściowego i wyjściowego po odpowiednich przełącznikach (odpowiednio: `-i` dla pliku wejściowego i `-o` dla pliku wyjściowego), np.

```
program -i przykladowyplikzgrafem.txt -o wynik
program -o wynik -i przykladowyplikzgrafem.txt
```

Pliki mogą nie posiadać rozszerzenia. Przełączniki mogą być podane w dowolnej kolejności. Uruchomienie programu bez żadnego parametru powoduje wyświetlenie komunikatu:

```
Uruchom program z parametrem -h aby dowiedziec sie wiecej
o poprawnej liscie parametrow
```

Uruchomienie programu z parametrem -h powoduje wyświetlenie krótkiej pomocy:

```
Parametry uruchomienia programu:
-h pomoc
-i plik wejsciowy
-o plik wyjsciowy
```

Uruchomienie programu z nieprawidłowymi parametrami powoduje wyświetlenie komunikatu

```
niewlasciwa lista parametrow
Uruchom program z parametrem -h aby dowiedziec sie wiecej o
poprawnej liscie parametrow
```

4 Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym.

4.1 Ogólna struktura programu

W funkcji głównej sprawdzane jest, czy plik wejściowy jest możliwy do odczytu, jeśli nie jest następuje wyświetlenie odpowiedniego komunikatu i program się kończy. Następnie wczytywane są kolejne wartości (identyfikatory wierzchołków) wierszami (krawędzie grafu) z pliku wejściowego. Każdy identyfikator jest przekazany do funkcji pobierzWierzcholekNaPodstawieID, która

w pierwszej kolejności sprawdza czy istnieje już wskaźnik do wierzchołka z takim samym co przekazanym identyfikatorem w wektorze wierzcholki . Jeśli taki już istnieje to zwraca ten wskaźnik, jeśli nie to alokuje nową strukturę, atrybutowi id przypisuje przekazany identyfikator, atrybutowi kolor przypisuje wartość zero (koloruje wierzchołek na szaro) i tworzy do niego wskaźnik, który później dodaje do wektora wierzcholki i go zwraca. Następnie następuje dodanie wskaźnika wskazującego na wierzchołek z końca krawędzi do wektora polaczenia wierzchołka z początku krawędzi oraz do wektora polaczenia wierzchołka z końca krawędzi dodawany jest wskaźnik, wskazujący na wierzchołek z początku krawędzi. Program w kolejnym kroku koloruje wierzchołek, na który wskazuje pierwszy wskaźnik znajdujący się w wektorze wierzcholki na czerwono, usuwa wskaźnik z wektora wierzcholki i dodaje go do wektora wierzcholkiPokolorowane. Następnie dla każdego wierzchołka znajdującego się w wektorze wierzcholkiPokolorowane zostaje wywołana funkcja pokolorujSasiednieWierzcholki , która koloruje wierzchołki połączone ze wskazywanym wierzchołkiem na kolor przeciwny od koloru wierzchołka, na który wskazuje przekazany wskaźnik. Następnie każdy wskaźnik do wierzchołka, który nie koloru szarego zostaje usunięty z wektora wierzcholki i dodany do wektora wierzcholkiPokolorowane. Po usunięciu wszystkich wierzchołków z wektora wierzcholki - czyli po pokolorowaniu wszystkich wierzchołków program wywołuje funkcję czyDwudzielny, która sprawdza przy wykorzystaniu funkcji czyDwudzielnyDlaWierzcholka czy każdy wierzchołek w grafie jest połączony tylko z wierzchołkami o przeciwnym kolorze do jego koloru. Gdy funkcja zwraca wartość **true**, jest to równoznaczne z dwudzielnością grafu zapisanego w pliku wejściowym; wartość **false** oznacza nie dwudzielność grafu. Program wypisuje do pliku wyjściowego zadany graf i komunikat, czy jest to graf dwudzielny, czy nie. Jeżeli zadany graf jest dwudzielny, program wypisuje wierzchołki obu grup (kolorów) grafu.

4.2 Szczegółowy opis typów i funkcji

Szczegółowy opis typów i funkcji zawarty jest w dodatku.

5 Testowanie

Program został przetestowany na różnego rodzaju plikach. Plikach poprawnych i typowych (w każdym wierszu znajdują się dwie liczby całkowite, bez zbędnych białych znaków oraz bez pustych wierszy). Plikach poprawnych ale nietypowych (w każdym wierszu znajdują się dwie liczby całkowicie, ale występują dodatkowe białe znaki oraz puste wiersze. Finalnie na plikach

niepoprawnych (plikach pustych lub z literami czy słowami zamiast liczb). Program został sprawdzony pod katem wycieków pamięci.

6 Wnioski

Program do sprawdzania dwudzielności grafu nieskierowanego jest programem stosunkowo prostym, chociaż wymaga podstawowej znajomości teorii grafów.Wykorzystuje proste struktyry i nieskomplikowane przejścia. Najbardziej wymagające okazało się odpowiednie wczytanie danych oraz wykorzystanie do tego odpowiednich struktur przechowywujące te dane. Nie do prostych zadań należało przystosowanie projektu do urochamiania go z linii poleceń z potrzebnymi przełącznikami. Wykonanie projektu pozwolało na sprawdzenie i doskonalenie nabytych umiejętności na laboratoriach oraz zdobyciu nowych.

7 Literatura

- "Wprowadzenie do teorii grafów", Robin J. Wilson

Dodatek
Szczegółowy opis typów i funkcji

Dwudzielny

Wygenerowano przez Doxygen 1.8.20

Rozdział 1

Indeks klas

1.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Rozdział 2

Indeks plików

2.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja struktury wierzcholek

`#include <struktury.h>`

Atrybuty publiczne

- int **id**
- int **kolor**
- vector< wierzcholek * > **polaczenia**

3.1.1 Opis szczegółowy

Struktura opisujaca kazdy wczytany wierzcholek

Parametry

id	przechowuje identyfikator wierzcholka
kolor	to kolor danego weierzcholka, przyjmuje wartosc 0 dla koloru szarego, 1 dla czerwonego i -1 dla niebieskiego
polaczenia	wektor przechowujacy wskazniki do wierzcholkow, z ktorymi dany wierzcholek jest polaczony

Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku Dwudzielny/Dwudzielny.cpp

```
#include "funkcje.h"
```

Funkcje

- int **main** (int iloscParametrow, char ∗parametry[ ])

4.2 Dokumentacja pliku Dwudzielny/funkcje.cpp

```
#include "funkcje.h"
```

Funkcje

- wierzcholek ∗ pobierzWierzcholekNaPodstawieID (vector< wierzcholek ∗ > &wierzcholki, int ID)
- void pokolorujSasiednieWierzcholki (wierzcholek ∗w)
- bool czyDwudzielnyDlaWierzcholka (wierzcholek ∗w)
- bool czyDwudzielny (vector< wierzcholek ∗ > &wierzcholkiPokolorowane)
- bool porownaj (char ∗text, string s2)

4.2.1 Dokumentacja funkcji

4.2.1.1 czyDwudzielny()

```
bool czyDwudzielny (
            vector< wierzcholek * > & wierzcholkiPokolorowane )
```

Funkcja sprawdza czy wszystkie wierzcholki sa poloczone z wierzcholkami o przeciwnym kolorze.

Funkcja sprawdza czy wszystkie wierzcholki w wektorze ze wskaznikami do pokolorowanych wierzcholkow maja wartosc true, nadawana przez funkcje 'czyDwudzielnyDlaWierzcholka'.

Parametry

wierzcholkiPokolorowane	wektor przechowujacy wskazniki do pokolorowanych juz wierzcholkow

4.2.1.2 czyDwudzielnyDlaWierzcholka()

```
bool czyDwudzielnyDlaWierzcholka (
            wierzcholek * w )
```

Funkcja sprawdza czy wskazywany wierzcholek jest polaczony z wierzcholkami o przeciwnym kolorze

W funkcji najpierw jest definiowany kolor jaki powinny miec wierzcholki ktore sa polaczone z wierzcholkiem, na ktory wskazuje przekazany wskaznik. Nastepnie w sprawdzane sa kolory wszystkich tych wierzcholkow. Jesli wszystkie kolory wiercholkow, polaczonych ze wskazywanym wierzcholkiem sa rowne 'kolorDoSasiadow' to zwracana jest wartosc true.

Parametry

w	wskaznik wskazujacy na sprawdzany wiercholek

Zwraca

odpowiedz czy jest dwudzielny dla wierzcholka

4.2.1.3 pobierzWierzcholekNaPodstawieID()

```
wierzcholek* pobierzWierzcholekNaPodstawieID (
            vector< wierzcholek * > & wierzcholki,
            int ID )
```

Funkcja najpierw sprawdza czy w wektorze przechowujacym wskazniki do poszczegolnych wierzcholkow, juz istnieje wierzcholek o podanym identyfikatorze. Jesli taki nie istnieje to tworzy nowy wierzcholek o przekaznym identyfikatorze.

Parametry

wierzcholki	wektor przechowujacy wskazniki do wierzcholkow
ID	id danego wierzcholka

Zwraca

wskaznik wskazujacy na wierzcholek o przekazanym ID

4.2.1.4 pokolorujSasiednieWierzcholki()

```
void pokolorujSasiednieWierzcholki (
            wierzcholek * w )
```

Funkcja koloruje wierzcholki danego wierzcholka na kolor przeciwny od koloru wierzcholka, na ktory wskazuje przekazny wskaznik.

Funkcja najpierw sprawdza czy kolor wierzcholka, na ktory wskazuje przekazany wskaznik jest rowny zero czyli szary. Kolor musi byc inny niz szary by moc przypisac wierzcholkom, ktore sa polaczone z wierzcholkiem, na ktory wskazuje przekazany wskaznik, kolor przeciwny do koloru wierzcholka, na ktory wskazuje przekazany wskaznik. Nastepnie definiujemy kolor, na ktory beda kolorowane wierzcholki znajdujace sie w wektorze polaczenia. W kolejnym kroku wszytskie wierzcholki znajdujace sie w owym wektorze sa na ten kolor kolorowane.

Parametry

w	wskaznik wskazujacy na dany wierzcholek

4.2.1.5 porownaj()

```
bool porownaj (
            char * text,
            string s2 )
```

Funkcja najpierw tworzy zmienna typu string, a nastepnie porownuje ja z drugim parametrem, ktory tez jest typu string.

Parametry

text	
s2	parametr typu string do ktorego porownujemy text

Zwraca

wartosc true gdy s1 jest rowne s2

4.3 Dokumentacja pliku Dwudzielny/funkcje.h

```
#include "struktury.h"
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include <string>
```

Funkcje

- wierzcholek * pobierzWierzcholekNaPodstawieID (vector< wierzcholek * > &wierzcholki, int ID)
- void pokolorujSasiednieWierzcholki (wierzcholek *w)
- bool czyDwudzielnyDlaWierzcholka (wierzcholek *w)
- bool czyDwudzielny (vector< wierzcholek * > &wierzcholkiPokolorowane)
- bool porownaj (char *text, string s2)

4.3.1 Dokumentacja funkcji

4.3.1.1 czyDwudzielny()

```
bool czyDwudzielny (
            vector< wierzcholek * > & wierzcholkiPokolorowane )
```

Funkcja sprawdza czy wszystkie wierzcholki sa poloczone z wierzcholkami o przeciwnym kolorze.

Funkcja sprawdza czy wszystkie wierzcholki w wektorze ze wskaznikami do pokolorowanych wierzcholkow maja wartosc true, nadawana przez funkcje 'czyDwudzielnyDlaWierzcholka'.

Parametry

wierzcholkiPokolorowane	wektor przechowujacy wskazniki do pokolorowanych juz wierzcholkow

4.3.1.2 czyDwudzielnyDlaWierzcholka()

```
bool czyDwudzielnyDlaWierzcholka (
            wierzcholek * w )
```

Funkcja sprawdza czy wskazywany wierzcholek jest polaczony z wierzcholkami o przeciwnym kolorze

W funkcji najpierw jest definiowany kolor jaki powinny miec wierzcholki ktore sa polaczone z wierzcholkiem, na ktory wskazuje przekazany wskaznik. Nastepnie w sprawdzane sa kolory wszystkich tych wierzcholkow. Jesli wszystkie kolory wiercholkow, polaczonych ze wskazywanym wierzcholkiem sa rowne 'kolorDoSasiadow' to zwracana jest wartosc true.

Parametry

w	wskaznik wskazujacy na sprawdzany wiercholek

Zwraca

odpowiedz czy jest dwudzielny dla wierzcholka

4.3.1.3 pobierzWierzcholekNaPodstawieID()

```
wierzcholek* pobierzWierzcholekNaPodstawieID (
             vector< wierzcholek * > & wierzcholki,
             int ID )
```

Funkcja najpierw sprawdza czy w wektorze przechowujacym wskazniki do poszczegolnych wierzcholkow, juz istnieje wierzcholek o podanym identyfikatorze. Jesli taki nie istnieje to tworzy nowy wierzcholek o przekaznym identyfikatorze.

Parametry

wierzcholki	wektor przechowujacy wskazniki do wierzcholkow
ID	id danego wierzcholka

Zwraca

wskaznik wskazujacy na wierzcholek o przekazanym ID

4.3.1.4 pokolorujSasiednieWierzcholki()

```
void pokolorujSasiednieWierzcholki (
             wierzcholek * w )
```

Funkcja koloruje wierzcholki danego wierzcholka na kolor przeciwny od koloru wierzcholka, na ktory wskazuje przekazny wskaznik.

Funkcja najpierw sprawdza czy kolor wierzcholka, na ktory wskazuje przekazany wskaznik jest rowny zero czyli szary. Kolor musi byc inny niz szary by moc przypisac wierzcholkom, ktore sa polaczone z wierzcholkiem, na ktory wskazuje przekazany wskaznik, kolor przeciwny do koloru wierzcholka, na ktory wskazuje przekazany wskaznik. Nastepnie definiujemy kolor, na ktory beda kolorowane wierzcholki znajdujace sie w wektorze polaczenia. W kolejnym kroku wszytskie wierzcholki znajdujace sie w owym wektorze sa na ten kolor kolorowane.

Parametry

w	wskaznik wskazujacy na dany wierzcholek

4.3.1.5 porownaj()

```
bool porownaj (
             char * text,
             string s2 )
```

Funkcja najpierw tworzy zmienna typu string, a nastepnie porownuje ja z drugim parametrem, ktory tez jest typu string.

Parametry

text	
s2	parametr typu string do ktorego porownujemy text

Zwraca

wartosc true gdy s1 jest rowne s2

4.4 Dokumentacja pliku Dwudzielny/struktury.h

```
#include <vector>
```

Komponenty

- struct wierzcholek

Indeks