Wstęp do programowania

Zadanie I

Paweł Daniluk

grudzień 2014

1 Grafy

Graf $G = (V, E)$ jest parą uporządkowaną, której elementami są:

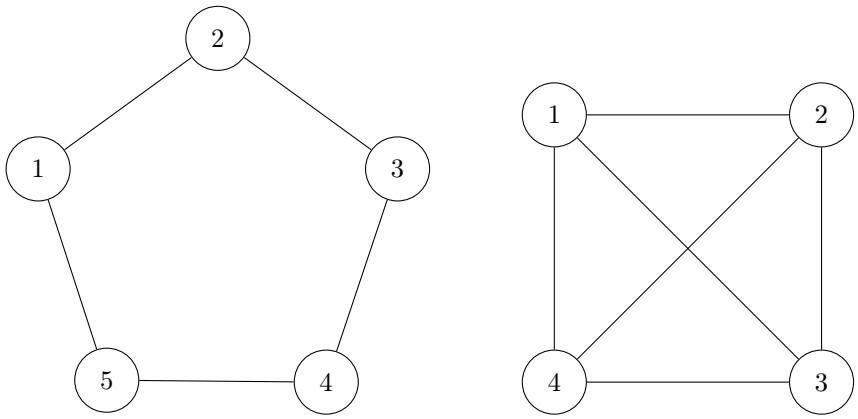
- V – zbiór wierzchołków grafu,
- $E \subseteq V \times V$ – zbiór krawędzi grafu.

Tak zdefiniowany obiekt może służyć do reprezentowania wszelkiego rodzaju zależności. Przykładem może być graf znajomości lub pokrewieństwa, w którym wierzchołki odpowiadają osobom, a krawędzie występują pomiędzy osobami, które się znają. Własności matematyczne grafów zostały bardzo dobrze przebadane. Wiele również wiadomo o złożoności obliczeniowej i optymalnych algorytmach rozwiązywania problemów zdefiniowanych przy ich pomocy. W szczególności rozważa się najkrótsze ścieżki pomiędzy wierzchołkami, cykle, kolorowania wierzchołków lub krawędzi, planarność (możliwość wykreślenia grafu bez przecinania się krawędzi), maksymalne kliki (podgrafy pełne), minimalne rozcięcia, izomorfizmy podgrafów i wiele innych aspektów.

Wyróżniamy dwa rodzaje grafów: skierowane i nieskierowane. W grafach skierowanych istotna jest kolejność wierzchołków w parach reprezentujących krawędzie. Mogą one służyć np. do reprezentowania zależności genealogicznych, albo podległości służbowej. W reprezentacji graficznej grafów skierowanych krawędzie oznacza się przy pomocy strzałek. Z kolei w grafach nieskierowanych kolejność wierzchołków w krawędzi nie jest istotna (krawędź (v_1, v_2) jest tożsama z (v_2, v_1)). W dalszym ciągu zajmować będziemy się wyłącznie grafami nieskierowanymi.

2 Reprezentacja grafu

Przyjmijmy dla ustalenia uwagi, że graf ma N wierzchołków, które ponumerowane są liczbami naturalnymi od 1 do N (czyli $V = \{1, \ldots, N\}$). Zauważmy, że przy takim założeniu, aby jednoznacznie określić graf, wystarczy podać zbiór E. Wyróżnimy trzy sposoby reprezentowania grafu w pamięci komputera.



(a) 5-cykl (b) 4-klika

Rysunek 1: Przykładowe grafy

2.1 Lista krawędzi

Jest to najłatwiejsza w implementacji i najmniej użyteczna metoda reprezentowania grafu. Przechowujemy zbiór E jako listę par lub tablicę dwuelementowych wektorów. Przykład:

```
[(1,2),(2,3),(3,4),(4,5),(5,1)] # 5-elementowy cykl
[(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)] # 4-elementowa klika
```

2.2 Listy sąsiedztwa

W tej reprezentacji z każdym wierzchołkiem związana jest lista jego sąsiadów (czyli wierzchołków, z którymi łączą go krawędzie). Jest to wygodna reprezentacja, jeżeli chcemy móc szybko dostać się do wielu sąsiadów wskazanego wierzchołka. Przykład:

```
[
  [2,5],     # wierzcholek 1
  [1,3],     # wierzcholek 2
  [2,4],     # wierzcholek 3
  [3,5],     # wierzcholek 4
  [1,4]      # wierzcholek 5
] # 5-elementowy cykl

[
  [2,3,4],     # wierzcholek 1
  [1,3,4],     # wierzcholek 2
  [1,2,4],     # wierzcholek 3
  [1,2,3]      # wierzcholek 4
```

```
] # 4-elementowa klika
```

2.3 Macierz sąsiedztwa

Macierzą sąsiedztwa grafu mającego N wierzchołków nazywamy kwadratową zero-jedynkową macierz $A \in \{0,1\}_{N,N}$ o rozmiarze N. Element a_{ij} tej macierzy odpowiada krawędzi pomiędzy wierzchołkami i oraz j i ma wartość 1, jeżeli są one połączone krawędzią i 0 w przeciwnym przypadku. Łatwo zauważyć, że jeżeli graf jest nieskierowany, jego macierz sąsiedztwa jest symetryczna. Przykład:

```
[
  [0,1,0,0,1],    # wierzcholek 1
  [1,0,1,0,0],    # wierzcholek 2
  [0,1,0,1,0],    # wierzcholek 3
  [0,0,1,0,1],    # wierzcholek 4
  [1,0,0,1,0]     # wierzcholek 5
] # 5-elementowy cykl

[
  [0,1,1,1],    # wierzcholek 1
  [1,0,1,1],    # wierzcholek 2
  [1,1,0,1],    # wierzcholek 3
  [1,1,1,0]     # wierzcholek 4
] # 4-elementowa klika
```

3 Problem

Celem zadania jest napisanie programu służącego do konwersji pomiędzy reprezentacjami grafu zapisanymi w plikach tekstowych. Program ma być uruchamiany z linii komend (graficzny interfejs użytkownika nie jest konieczny) i przyjmować następujące obowiązkowe argumenty:

`-in-edges|-in-lists|-in-matrix` wzajemnie wykluczające się argumenty określające typ pliku wejściowego

`-in=`**plik** nazwa pliku wejściowego

`-out-edges|-out-lists|-out-matrix` wzajemnie wykluczające się argumenty określające typ pliku wyjściowego

`-out=`**plik** nazwa pliku wyjściowego (podanie `-` spowoduje wypisanie wyniku na standardowe wyjście)

Przykładowe wywołanie:

```
$ graph_conv.py --in-lists --in=wejscie.txt --out-matrix --out=-
```

4 Formaty plików

Pierwsza linia pliku w każdym z formatów zawiera liczbę wierzchołków grafu. Poniżej pliki zawierające grafy z powyższych przykładów.

4.1 Lista krawędzi

5-cykl:

```
5
1 2
2 3
3 4
4 5
5 1
```

4-klika:

```
4
1 2
1 3
1 4
2 3
2 4
3 4
```

4.2 Listy sąsiedztwa

5-cykl:

```
5
2 5
1 3
2 4
3 5
1 4
```

4-klika:

```
4
2 3 4
1 3 4
1 2 4
1 2 3
```

4.3 Macierz sąsiedztwa

5-cykl:

```
5
0 1 0 0 1
1 0 1 0 0
0 1 0 1 0
0 0 1 0 1
1 0 0 1 0
```

4-klika:

```
4
0 1 1 1
1 0 1 1
1 1 0 1
1 1 1 0
```