

	0	1	2	3	4	5	6	7	8	9
0	0	1	0	0	0	0	1	0	1	0
1	1	0	0	0	1	0	1	0	0	1
2	0	0	0	0	1	0	1	0	0	0
3	0	0	0	0	1	1	0	0	1	0
4	0	1	1	1	0	1	0	0	0	1
5	0	0	0	1	1	0	0	0	0	0
6	1	1	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	1
8	1	0	0	1	0	0	0	1	0	0
9	0	1	0	0	1	0	0	1	0	0

W ukazanej przykładowej macierzy sąsiedztwa widzimy, że jedynka jest także na indeksie [5, 3], a zatem te punkty łączy krawędź obustronna.

Jeśli przyjrzymy się dokładniej, zauważymy że macierz ta jest symetryczna. Oznacza to, że reprezentuje ona graf nieskierowany i wszystkie jego krawędzie są obustronne.

2. Omówienie algorytmów

Podczas zajęć poznaliśmy dwa algorytmy służące do generowania grafów, a konkretnie ich macierzy sąsiedztwa.

Pierwszy z nich działa na zasadzie losowania koordynatów komórki w macierzy sąsiedztwa, następnie sprawdzenia czy znajduje się na nich 0 i jeśli tak, wstawia na jego miejsce jedynkę, lub dla grafu ważonego losowo wybraną wagę. Jeśli wylosowana komórka była już wypełniona, algorytm losuje koordynaty jeszcze raz. Cały proces powtarza się, dopóki nie zostanie utworzona zadana przez użytkownika liczba krawędzi.

Pseudokod:

Powtarzaj:

 Powtarzaj:

 Losuj „x” z przedziału od 0 do n

 Losuj „y” z przedziału od 0 do n

 Dopóki macierz[x, y] != 0

 Jeśli typ grafu to ważony:

 Losuj „waga” z przedziału od 1 do 9 // Zamiast 9 może być dowolna inna wartość

 W przeciwnym wypadku

 waga = 1

 macierz[x, y] = waga

 Jeśli typ grafu to nieskierowany:

 macierz[y, x] = waga

 iloscWygenerowanych++

Dopóki iloscWygenerowanych < iloscDoWygenerowania

Gdzie „n” to zadana ilość węzłów grafu.

Przykładowa macierz wygenerowana przez moją implementację algorytmu:

```

Lukasz Konieczny LK4 LAB4
Program generujacy macierze sasiedztwa grafow
MENU:
1. Algorytm losowego tworzenia grafu
2. Algorytm R-MAT
Wpisz swoj wybor: 1
Podaj ilosc wezlow: 8
Podaj ilosc krawedzi: 6
Czy graf ma byc skierowany? Wpisz 1 jesli tak: 1
Czy graf ma byc wazony? Wpisz 1 jesli tak: 0

1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0
0 1 0 0 0 0 1 0
0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1

```

Kolejnym przedstawionym nam algorytmem był algorytm R-MAT (recursive matrix). Służy on do generowania grafów maksymalnie zbliżonych do realnych modeli relacji. Jest w stanie to osiągnąć poprzez nietypowe podejście do wyboru komórek w macierzy sąsiedztwa.

Na początku tworzy kwadratową macierz sąsiedztwa wielkości 2^n , gdzie „n” podaje użytkownik. Następnie, na bazie podanej przez użytkownika oczekiwanej „gęstości” grafu (czyli stosunku liczby krawędzi do liczby wszystkich możliwych do utworzenia krawędzi) wyznacza, ile krawędzi powinien wygenerować.

Następnie dzieli macierz na ćwiartki.

Operując na koordynatach x, y oraz p, gdzie „p” oznacza rozmiar aktualnie wylosowanej ćwiartki, losuje jedną z nich zgodnie z podanym przez użytkownika prawdopodobieństwem (suma prawdopodobieństw wylosowania każdej ćwiartki powinna wynosić 100%) i odpowiednio zwiększa x oraz y, dopóki p nie jest równe 1, co oznacza, że nie jesteśmy w stanie stworzyć mniejszej ćwiartki i wybraliśmy jedną komórkę macierzy.

Następnie ustawiamy wartość tej komórki zgodnie z typem generowanego grafu. Operację losowania ćwiartki powtarzamy aż do uzyskania oczekiwanej liczby krawędzi.

Pseudokod:

```

rozmiar = 2n
iloscKrawedzi = gestosc / 100 * rozmiar * rozmiar
p = rozmiar
Powtarzaj:
    x = 0
    y = 0
    Powtarzaj:
        p = p / 2
        Losuj „wyborCwiartki” z przedziału od 1 do 100
        Jeśli wylosowano ćwiartkę a:
            // Nie rób nic
        Inaczej, jeśli wylosowano ćw. b:
            x = x + p
        Inaczej, jeśli wylosowano ćw. c:
            y = y + p
        W przeciwnym wypadku wylosowano ćw. d:
            x = x + p
            y = y + p
    Dopóki „p” różne od 1

    Jeśli macierz[x][y] == 0:
        macierz[x][y] = 1
        aktualnalloscKrawedzi++
        Jeśli graf jest nieskierowany:
            macierz[y][x] = 1
    W przeciwnym wypadku:
        Jeśli graf jest ważony:
            macierz[x][y]++
            Jeśli dodatkowo jest nieskierowany i „x” różne od „y”:
                macierz[y][x]++

Dopóki aktualnalloscKrawedzi < iloscKrawedzi

```

Tak wygląda przykładowa macierz sąsiedztwa, wygenerowana przez moją implementację algorytmu R-MAT:

```

Lukasz Konieczny LK4 LAB4
Program generujący macierze sąsiedztwa grafów
MENU:
1. Algorytm losowego tworzenia grafu
2. Algorytm R-MAT
Wpisz swój wybór: 2
Podaj N: 3
Podaj gęstość (w procentach!): 30
Podaj prawdopodobieństwo ćwiartki a (w procentach!): 20
Podaj prawdopodobieństwo ćwiartki b (w procentach!): 30
Podaj prawdopodobieństwo ćwiartki c (w procentach!): 15
Na bazie wprowadzonych danych, prawdopodobieństwo d wynosi 35%
Czy graf ma być skierowany? Wpisz 1 jeśli tak: 1
Czy graf ma być ważony? Wpisz 1 jeśli tak: 0

1 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 1 0 0 0 0
0 1 0 0 0 1 0 0
0 0 1 0 0 0 0 0
0 1 0 0 0 0 1 0
0 1 1 0 0 1 1 1
0 0 1 1 0 1 1 0

```

3. Wnioski

Podczas zajęć dowiedzieliśmy się czym jest graf, jakie są jego warianty oraz zaimplementowaliśmy dwa algorytmy pozwalające wygodnie generować nasze własne grafy (a konkretnie ich macierze sąsiedztwa). Poznane algorytmy tracą na wydajności poprzez element losowego wybierania koordynatów komórek w macierzy sąsiedztwa, co może doprowadzić do wielokrotnego wyboru tej samej komórki i spowodować program, jednak z reguły ich złożoność czasowa zależy od ilości węzłów i oczekiwanej ilości krawędzi (lub gęstości) generowanego grafu.

Bibliografia

- Materiały z wykładu o strukturach danych
- Opis zadania na platformie Delta
- Deepayan Chakrabarti, Yiping Zhan, and Christos Faloutsos, R-MAT: A Recursive Model for Graph Mining, Proceedings of the 2004 SIAM International Conference on Data Mining
- Opis algorytmu R-MAT i jego pseudokod zamieszczony przez użytkownika „kubpica” w serwisie GitHub [kubpica/GeneracjaGrafowR-MAT: Algorytm generujący graf o wymiarach \$n^2\$ i podanej przez użytkownika gęstości. \(github.com\)](https://github.com/kubpica/GeneracjaGrafowR-MAT)