

- **Opis problemu:**

Graf – to struktura danych składająca się z wierzchołków i krawędzi, przy czym poszczególne wierzchołki (zwane też węzłami) mogą być połączone krawędziami (skierowanymi lub nieskierowanymi) w taki sposób, iż każda krawędź zaczyna się i kończy w którymś z wierzchołków.

Wierzchołki i krawędzie mogą być numerowane, etykietowane i zawierać pewną dodatkową informację - w zależności od potrzeby modelu, do którego konstrukcji są wykorzystane. W porównaniu do drzew w grafach mogą występować pętle i cykle. Krawędzie mogą mieć wyznaczony kierunek (wtedy graf nazywamy skierowanym), mogą mieć przypisaną wagę (pewną liczbę).

Graf **R-MAT (Recursive MATrix)** – jest **losowym** grafem z **rekurencyjnie** generowaną strukturą na podstawie **prawdopodobieństwa** wyboru danej krawędzi. Może posłużyć do modelowania wielu rzeczywistych problemów, sieć komputerowych czy symulacji. Parametrem dodatkowym którym możemy operować jest gęstość oraz rodzaj grafu.

- **Opis algorytmu:**

Algorytm przedstawia graf za pomocą macierzy sąsiedztwa o wymiarach $n=2^k$. Operuje na dynamicznej tablicy dwuwymiarowej wypełnionej zerami. Wierzchołek sąsiadujący sam ze sobą oznaczony został jedynką. W ten sposób każda tworzona macierz ma przekątną złożoną z jedynek.

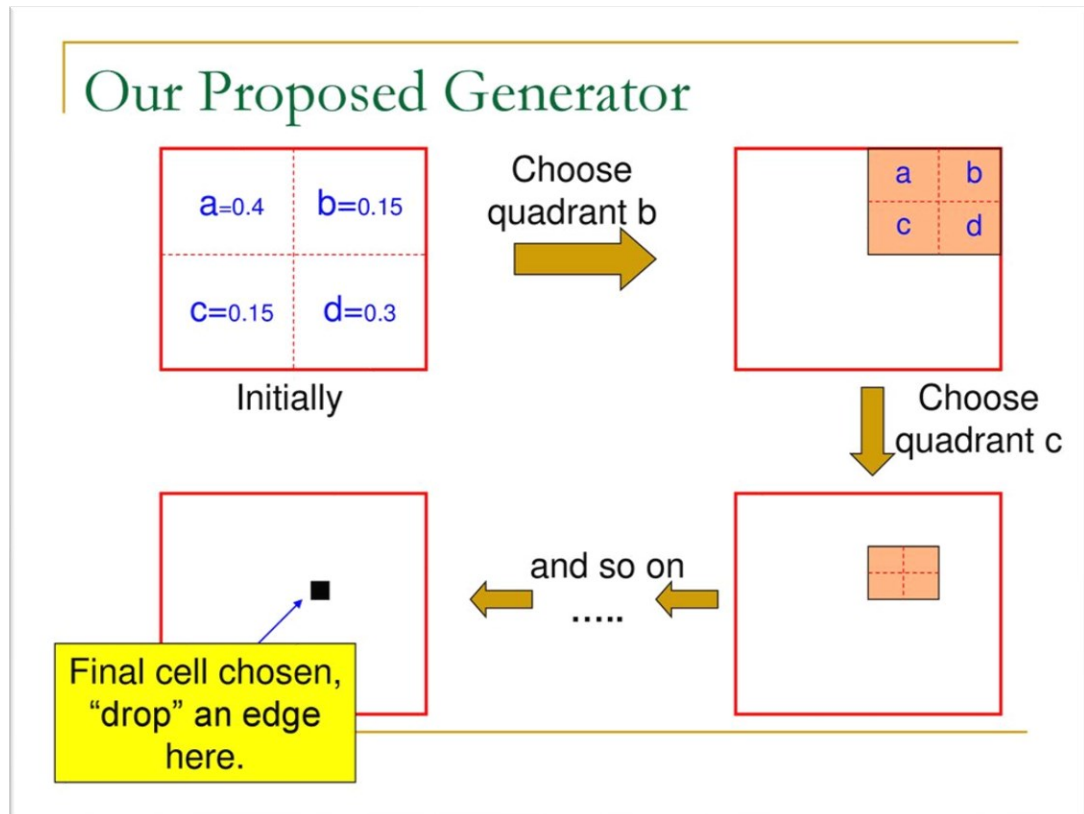
Generowanie macierzy grafu:

- Tworzę dynamiczną tablicę dwuwymiarową $tab[n][n]$ (gdzie $n = 2^k$) i wypełniam ją zerami,
- Przekątną macierzy wypełniam jedynkami (wierzchołki sąsiadują same ze sobą),
- Liczę ilość jedynek do wpisania na podstawie zadanej gęstości:
 $ilość_jedynek = pole_macierzy * gęstość / 100$;
W przypadku gdy graf jest nieskierowany dzielę wynik przez 2 gdyż obie połówki macierzy sąsiedztwa są symetryczne.
- Wywołuję funkcję $graph(x, y, size)$ tyle razy ile jest jedynek do wpisania, (Problem gęstości nie mógł zostać rozwiązany przy pomocy rekursji co zostało wyjaśnione we wnioskach.)

Funkcja $graph(x, y, s)$:

- Funkcja operuje na współrzędnych prawego dolnego rogu kwadratu o boku s .
- W przypadku gdy $s \neq 1$ następuje odwołanie do funkcji losującej liczby (0,1,2,3) z podanym wcześniej prawdopodobieństwem. Liczby te reprezentują poszczególne części kwadratu (a,b,c,d).
- Zależnie od wylosowanej części wywołuję rekurencyjnie funkcję zmieniając współrzędne prawego dolnego rogu kwadratu i dzieląc długość boku kwadratu na pół.
Jeżeli A to $graph(x-s/2, y-s/2, s/2)$;
Jeżeli B to $graph(x, y-s/2, s/2)$;
Jeżeli C to $graph(x-s/2, y, s/2)$;
Jeżeli D to $graph(x, y, s/2)$;

- W przypadku gdy $s=1$ ale $\text{tab}[x][y]$ jest już jedynką wywołuję funkcję `graph` ze początkowymi parametrami (`graph(n,n,n)`).
- W przypadku gdy $s=1$ i $\text{tab}[x][y]$ jest zerem, wstawiam w to miejsce jeden. Jeżeli graf jest nieskierowany należy również wstawić jedynkę w miejsce $\text{tab}[y][x]$.



Działanie algorytmu przedstawione w sposób graficzny

- **Fragment kodu:**

Funkcja wykorzystująca rekurencję oraz funkcja losująca z prawdopodobieństwem

```
int graph(int x, int y, int s)
{
    if(s!=1)
    {
        int l = lotto();
        if(l==0) graph(x-s/2, y-s/2, s/2);
        else if(l==1) graph(x, y-s/2, s/2);
        else if(l==2) graph(x-s/2, y, s/2);
        else graph(x, y, s/2);
    }
    else if(tab[x-1][y-1]==1) graph(n,n,n);
    else
    {
        if(r){tab[y-1][x-1]=1;};
        tab[x-1][y-1]=1;
    }
}

int lotto()
{
    int l = (rand() % 100) + 1;
    if(l<=a) return 0;
    else if(l<=a+b) return 1;
    else if(l<=a+b+c) return 2;
    else return 3;
}
```

Przykładowe wywołanie programu:

[illegible]

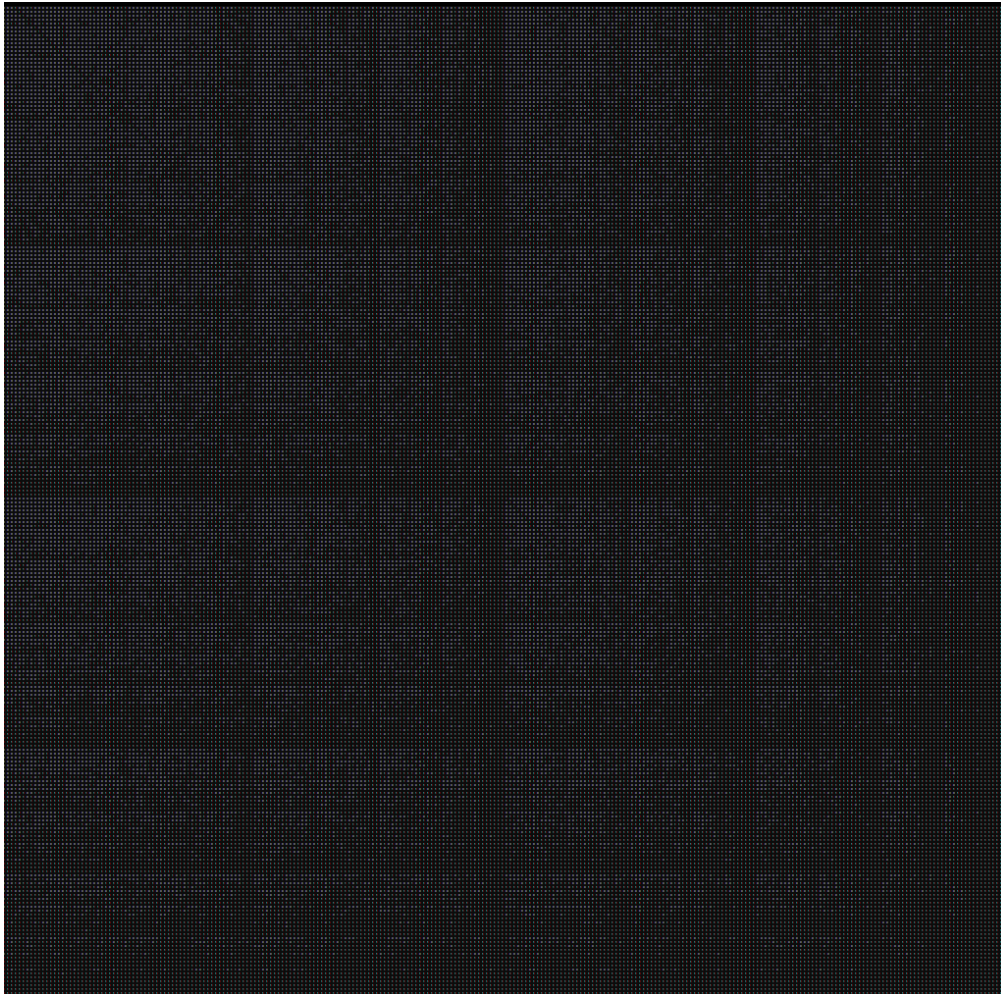
```
Generowanie macierzy grafu R-MAT
Rodzaj grafu (0-skierowany, 1-nieskierowany): 1
Ilosc wierzchołkow: n=2^k, podaj k: 5
Gestosc (w %): 60
Prawdopodobienstwo (w %, a+b+c+d=100)
a: 15
b: 15
c: 15
d: 55
```

```
Generowanie macierzy grafu R-MAT
Rodzaj grafu (0-skierowany, 1-nieskierowany): 0
Ilosc wierzchołkow: n=2^k, podaj k: 7
Gestosc (w %): 50
Prawdopodobienstwo (w %, a+b+c+d=100)
a: 10
b: 10
c: 10
d: 70
```

Ciekawy okresowy rozkład stopni wierzchołków grafów można zaobserwować przy dużych macierzach. Macierz dalej przedstawiona jest za pomocą 0 i 1 lecz została odpowiednio przeskalowana.

Mimo iż graf nie jest symetryczny względem przekątnej, można zauważyć pewne zależności między dwoma połówkami.

```
Generowanie macierzy grafu R-MAT
Rodzaj grafu (0-skierowany, 1-nieskierowany): 1
Ilosc wierzchołkow:  $n=2^k$ , podaj k: 8
Gęstość (w %): 65
Prawdopodobieństwo (w %,  $a+b+c+d=100$ )
a: 10
b: 40
c: 10
d: 40
```



Gdy przypatrzymy się grafice (szczególnie lewy górny róg) możemy łatwo zauważyć że graf jest nieskierowany (macierz jest symetryczna względem przekątnej).

- **Wnioski:**

Program poprawnie generuje grafy R-MAT. Poprzez manipulację parametrami takimi jak gęstość, prawdopodobieństwo czy rodzaj grafu możemy dowolnie wpływać na jego budowę. Warto również zauważyć, że o ile sam algorytm został zrealizowany przy pomocy rekurencji to problem gęstości już nie gdyż przy dużych rozmiarach macierzy (rzędu 2^9), kończyła się dostępna pamięć ram. Jest to wada rekursji która czasem uniemożliwia jej wykorzystanie.

Źródła:

- [1]. Deepayan Chakrabarti, Yiping Zhan, Christos Faloutsos R-MAT: A Recursive Model for Graph Mining
- [2]. PODSTAWY INFORMATYKI home.agh.edu.pl/~horzyk/lectures/pi/ahdydpiwykl9.html