

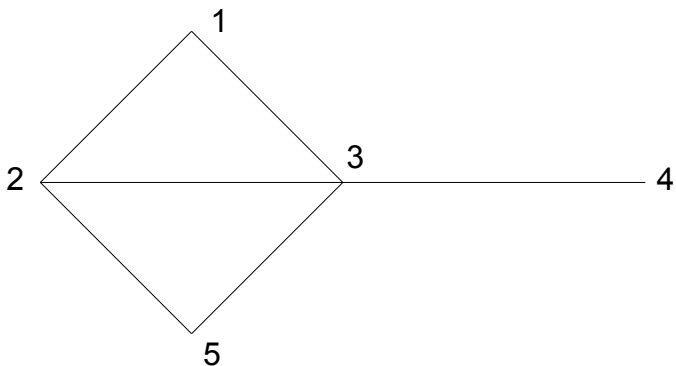
Grafy

- Graf (graf ogólny) to para $G(V,E)$, gdzie:
 - V jest zbiorem wierzchołków, (czasami zwanymi węzłami lub punktami grafu)
 - E jest rodziną (być może powtarzających się) krawędzi, czyli jedno- i dwu-elementowych podzbiorów V .
- Graf prosty to para $G(V,E)$, gdzie:
 - V jest zbiorem wierzchołków,
 - E jest zbiorem krawędzi między różnymi wierzchołkami, czyli dwu-elementowych podzbiorów V .

Macierz sąsiedztwa

- Macierz sąsiedztwa $A(G)$ grafu prostego $G=(\{v_1, \dots, v_n\}, E)$ to zero-jedynkowa macierz $\langle a_{ij} \rangle$ rozmiaru $n \times n$, gdzie

$$a_{ij} = \begin{cases} 1 & \text{jeżeli } v_i, v_j \in E \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$



$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Grafy

- Więcej o pojęciach związanych z grafami powiemy na „Matematyce dyskretnej”.

Przechodzenie grafu w głąb

- DFS – Depth First Search
- Startujemy z dowolnego wierzchołka. Odwiedzamy dowolnego nieodwiedzonego sąsiada. Z niego wchodzimy do kolejnego nieodwiedzonego sąsiada, itd.
- Wykorzystujemy rekurencję czyli stos LIFO.

Przechodzenie grafu w głąb

```
PROCEDURE DFS (wierzchołek v) ;  
BEGIN  
    zaznacz v jako odwiedzony ;  
    dla każdego nieodwiedzonego sąsiada n  
        DFS (n) ;  
END ;
```

Przechodzenie grafu wszerz

- BFS – Breadth First Search
- Startujemy z dowolnego wierzchołka. Odwiedzamy wszystkich nieodwiedzonych sąsiada. Następnie ich sąsiadów, itd.
- Wykorzystujemy kolejkę FIFO.

Przechodzenie grafu wszerz

```
PROCEDURE BFS;  
BEGIN  
    zaznacz wierzchołek startowy jako odwiedzony;  
    wrzuć wierzchołek startowy do kolejki;  
    WHILE kolejka nie jest pusta  
    BEGIN  
        pobierz wierzchołek v z kolejki;  
        dla wszystkich sąsiadów n  
        IF sąsiad n nieodwiedzony THEN  
            zaznacz n jako odwiedzony;  
            dodaj n do kolejki;  
        END;  
    END;  
END;  
END;
```

Rysowanie w C++ Builder

- Zdarzenie TForm::OnPaint
- Obiekt Canvas, metody MoveTo(), LineTo(), Ellipse(), itp.
- TForm->Invalidate() w zdarzeniu TForm::OnResize
- $x = \sin(2 * M_PI * i / illosc) * 100 + 200;$
- $y = \cos(2 * M_PI * i / illosc) * 100 + 200;$