

ASD, 2022.05.12, Wyk. 12

Algorytm grafowe.

Def

Grat $G = (V, R)$ to para uporządko-

wana taka, że

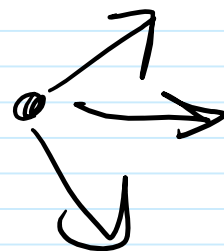
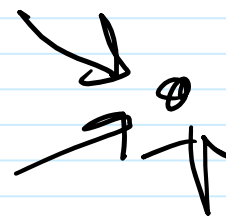
V - to niepusty zb. liczb

$R \in V \times V$ - zbiór krawędzi.

Jeśli $\forall a, b \in V$ ($aRb \rightarrow bRa$) to

graf G jest symetryczny (niekierowany).

stopień wchodzący, wienchodzący
wychodzący.



Jak reprezentować graf?
Jak zaimplementować graf?

Macierz sąsiedztwa (incydencji)

Wierz $V = \{0, \dots, N\}$ - zb. wierzchołków

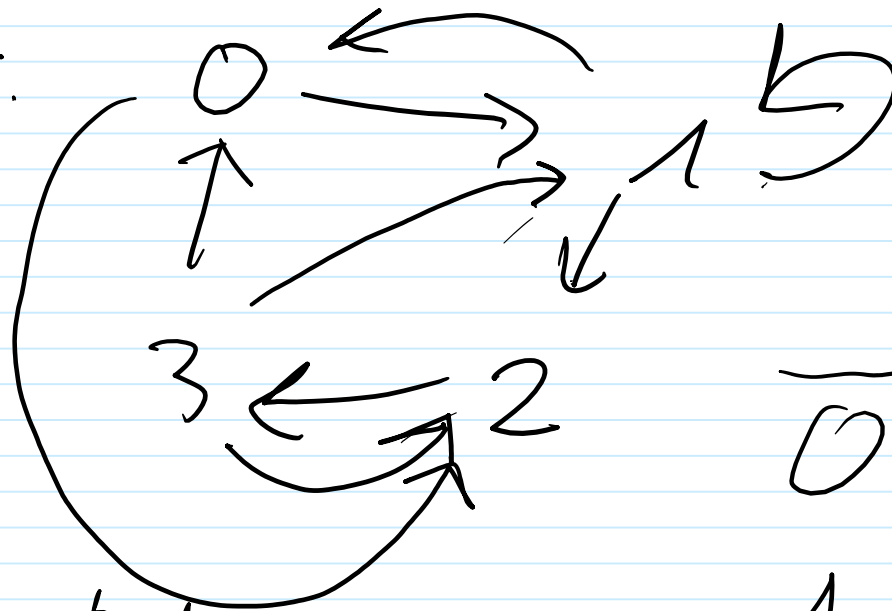
$$G = (V, E)$$

int $t_G[N+1][N+1]$ gdzie

$$t_G[i][j] = 1 \quad \text{wz} \quad (i, j) \in E$$

$$t_G[i][j] = 0 \quad \text{wz} \quad (i, j) \notin E$$

Up



zadaty: prostota
 reprezentacja macierzowa
 pozwala wykorzystać algorytm
 działający na macierzach
 do obliczenia własności grafów.

	0	1	2	3
0	0	1	1	0
1	1	1	1	0
2	0	0	1	1
3	1	1	1	0

Wady

Jedki grst ma mało krawędzi albo

bardzo dużo krawędzi to

małemu sąsiadstwu zajmuje dużo

miejsa a trzymamy w niej ^{prawe} same

zera lub prawie same jedynki.

Tablica list sąsiedztwa

$V = \{0, \dots, N\}$ - wierzchołki

$R \subseteq V \times V$

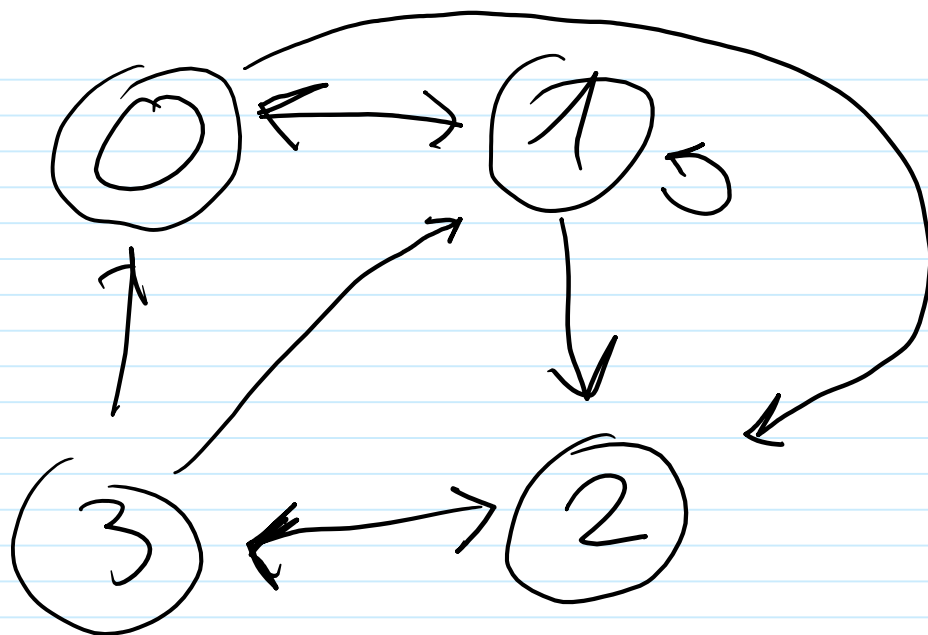
```
struct Edge {
```

```
    int end;
```

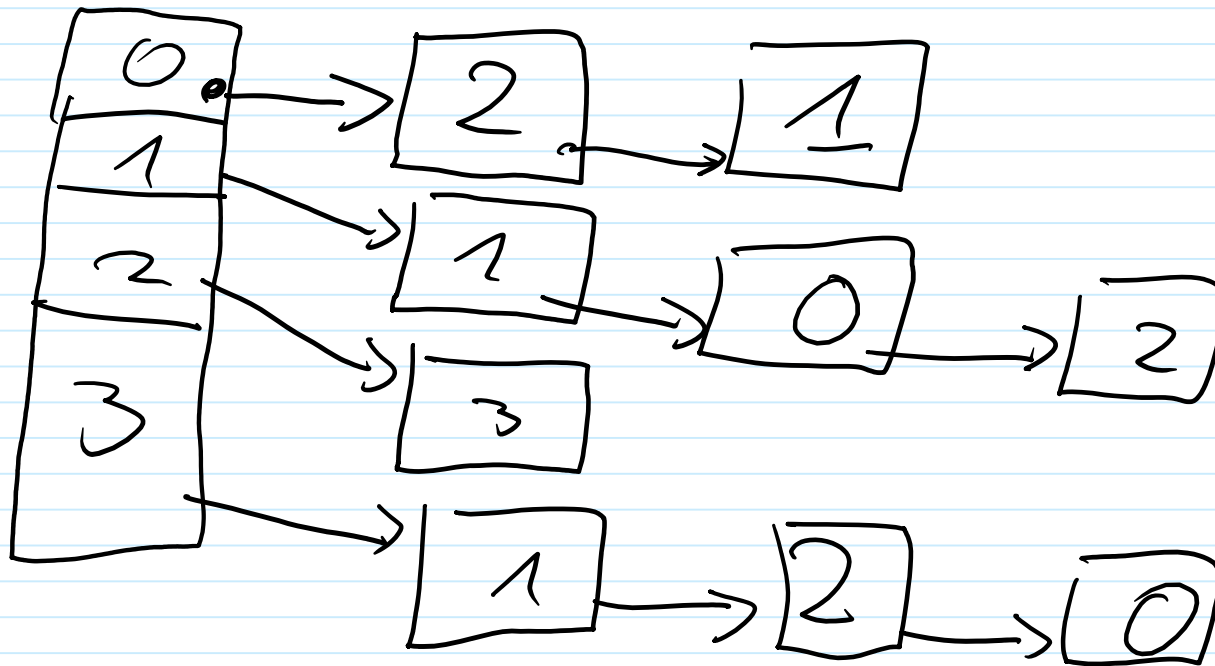
```
    Edge * next;
```

```
} Edge;
```

```
(Edge *) graf[0..N];
```



grat



Ta reprezentacja może być
bardziej efektywna pamięciowo,
jeśli w grafie nie ma dwóch krawędzi,
ale np. sprawdzenie czy jest krawędź
z wierzchołka i do wierzchołka;
zajmując tyle czasu jak i jest stopień
wierzchołka; (w macierzowej reprezentacji
musimy to zrobić w
czasie stałym).