

Algorytmy i struktury danych

Grafy I

IO, WiMiP

Danuta Szeliga

AGH Kraków

Spis treści

- 1 Graf
 - Implementacja podstawowych algorytmów grafowych

Reprezentacja grafu $G = (V, E)$

- *Listy sąsiedztwa*

- Dla grafów rzadkich, tzn. $|E| \ll |V|^2$
- Tablica o rozmiarze $|V|$ składająca się z list, po jednej dla każdego wierzchołka
- Złożoność pamięciowa $\Theta(|V| + |E|)$
- Wyszukiwanie krawędzi: przeglądanie list

- *Macierze sąsiedztwa*

- Dla grafów gęstych, tzn. $|E| \approx |V|^2$

-

$$m_{ij} = \begin{cases} 1 & \exists e = (v_i, v_j) \\ 0 & \text{wpp} \end{cases}$$

- Złożoność pamięciowa $\Theta(|V|^2)$
- Wyszukiwanie krawędzi $O(1)$

- *Macierz incydencji*

-

$$b_{ij} = \begin{cases} -1 & \text{jeżeli krawędź } j \text{ wychodzi z wężła } i \\ 1 & \text{jeżeli krawędź } j \text{ wchodzi do wężła } i \\ 0 & \text{wpp} \end{cases}$$

- Złożoność pamięciowa $\Theta(|V| \cdot |E|)$
- Wyszukiwanie krawędzi $O(1)$

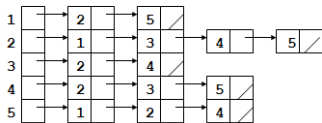
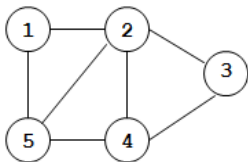
Reprezentacja grafów etykietowanych

Reprezentacja grafów etykietowanych (z wagami)

- Na liście sąsiedztwa waga krawędzi (v_i, v_j) jest pamiętana z wierzchołkiem
- W macierzy sąsiedztwa $m_{ij} = w(v_i, v_j)$

Reprezentacja grafów. Przykład

Graf nieskierowany G



	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

Lista sąsiedztwa grafu G

Macierz sąsiedztwa grafu G

Implementacja podstawowych algorytmów grafowych

Podstawowe algorytmy grafowe

- Przeszukiwane grafów wszerz
- Przeszukiwane grafów wgłąb

Przeszukiwanie grafu wszerz I

- Przeszukiwanie wszerz
 - Jeden z najprostszych algorytmów przeszukiwania grafu
 - Dany jest graf $G = (V, E)$ i jeden wyróżniony wierzchołek s (źródło). W przeszukiwaniu wszerz badane są krawędzie G w celu odwiedzenia każdego wierzchołka osiągalnego z s . Obliczane są też odległości (najmniejsza liczba krawędzi) od s do wszystkich osiągalnych wierzchołków
 - Wynik algorytmu: drzewo o korzeniu s , zawierające wszystkie wierzchołki osiągalne z s
 - Dla każdego wierzchołka v osiągalnego z s ścieżka w drzewie przeszukiwania od s do v jest równa najkrótszej ścieżce od s do v w grafie G

Przeszukiwanie grafu wszerz II

- Założenia algorytmu
 - W algorytmie wierzchołki są kolorowane na biało, szaro lub czarno
 - Podczas przeszukiwania każdy nowo napotkany wierzchołek staje się odwiedzony i zmienia kolor na szary lub czarny
 - Na początku wszystkie wierzchołki są białe
 - Jeśli $(u, v) \in E$ i wierzchołek u jest czarny, to v jest albo szary albo czarny, tzn. wszystkie wierzchołki sąsiadujące z v są już odwiedzone
 - Wierzchołki szare mogą mieć białych sąsiadów

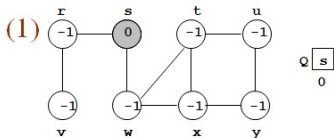
Algorytm przeszukiwania wszere grafów

- Dany jest graf $G = (V, E)$ reprezentowany przez listy sąsiedztwa. Dla każdego wierzchołka $u \in V$ dana jest lista sąsiedztwa $Ls[u]$
- Kolor wierzchołka $u \in V$ jest zapamiętany w zmiennej $color[u]$
- Poprzednik wierzchołka $u \in V$ jest zapamiętany w zmiennej $poprz[u]$. Jeśli u nie ma poprzednika, to $poprz[u] = -1$
- Odległość od źródła s do wierzchołka u jest obliczana w zmiennej $odl[u]$
- Szare wierzchołki są zapamiętywane w kolejce Q typu FIFO

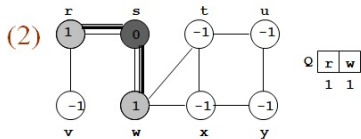
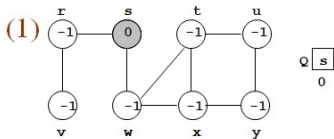
Algorytm przeszukiwania wszere grafów - cd

```
for(każdy wierzchołek  $u \in V[G] - \{s\}$ ){
    kolor[u] = biały;
    odl[u] = -1;
    poprz[u] = -1;
}
kolor[s] = szary; odl[s]= 0; poprz[s] = -1;
Q = {s};
while(Q !=  $\emptyset$ ){
    u = weź wierzchołek z kolejki Q;
    for(każdy wierzchołek  $v \in Ls[u]$ ){
        if(kolor[v]==biały){
            kolor[v]=szary;
            odl[v] = odl[u] + 1;
            poprz[v] = u;
            dodaj do kolejki Q wierzchołek v;
        }
    }
    kolor[u] = czarny;
}
```

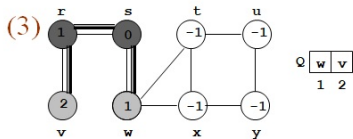
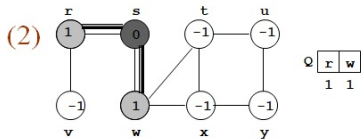
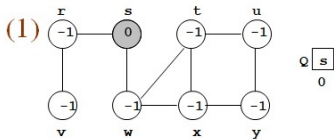
Algorytm przeszukiwania wszerz grafów. Przykład



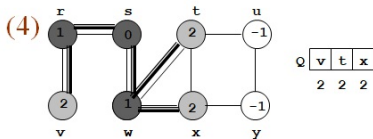
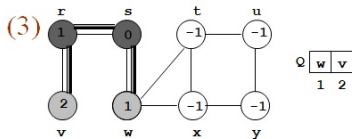
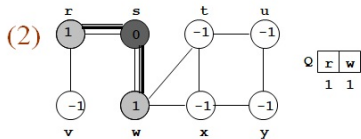
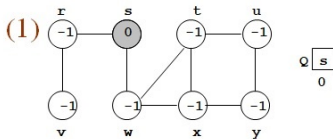
Algorytm przeszukiwania wszerz grafów. Przykład



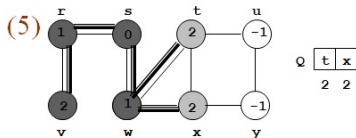
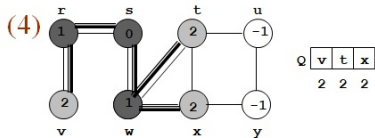
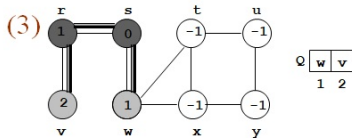
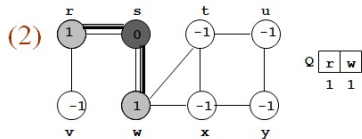
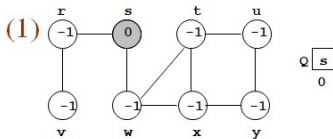
Algorytm przeszukiwania wszere grafów. Przykład



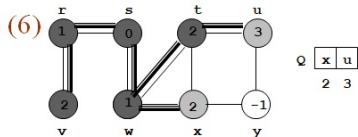
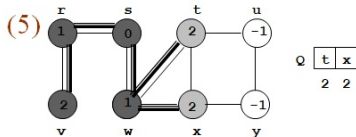
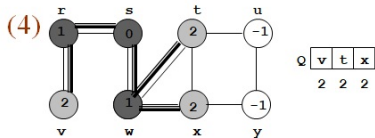
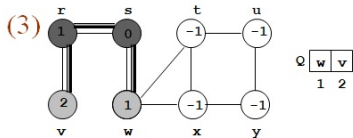
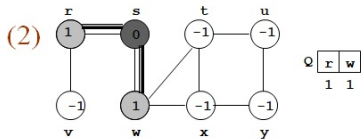
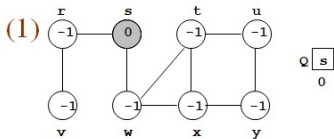
Algorytm przeszukiwania wszerz grafów. Przykład



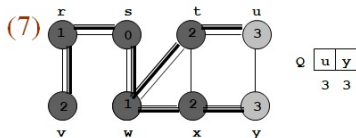
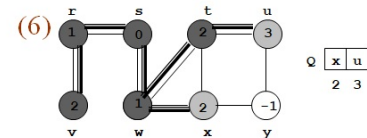
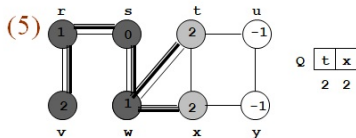
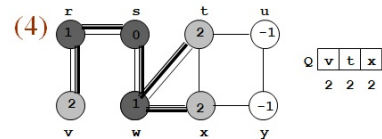
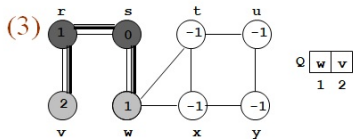
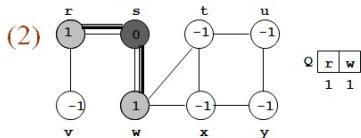
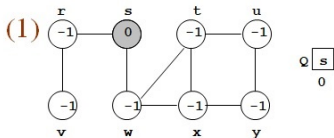
Algorytm przeszukiwania wszerz grafów. Przykład



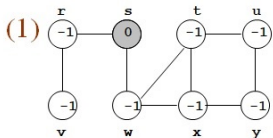
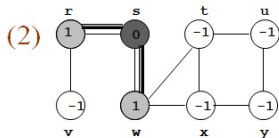
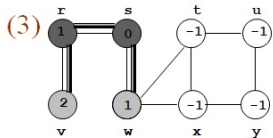
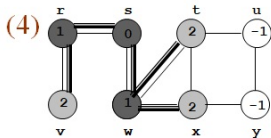
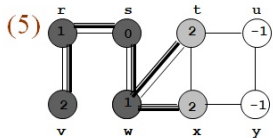
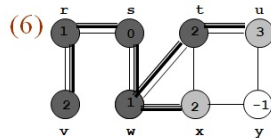
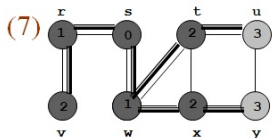
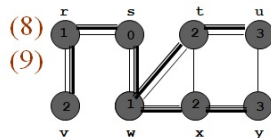
Algorytm przeszukiwania wszerz grafów. Przykład



Algorytm przeszukiwania wszerz grafów. Przykład



Algorytm przeszukiwania wszerz grafów. Przykład


 $Q \begin{bmatrix} s \\ 0 \end{bmatrix}$

 $Q \begin{bmatrix} r & w \\ 1 & 1 \end{bmatrix}$

 $Q \begin{bmatrix} w & v \\ 1 & 2 \end{bmatrix}$

 $Q \begin{bmatrix} v & t & x \\ 2 & 2 & 2 \end{bmatrix}$

 $Q \begin{bmatrix} t & x \\ 2 & 2 \end{bmatrix}$

 $Q \begin{bmatrix} x & u \\ 2 & 3 \end{bmatrix}$

 $Q \begin{bmatrix} u & y \\ 3 & 3 \end{bmatrix}$


(9) $Q = \emptyset$

Przeszukiwane wgłąb grafów

- Przy przeszukiwaniu wgłąb badane są krawędzie ostatnio odwiedzonego wierzchołka v , z którego jeszcze wychodzą nie zbadane krawędzie
- Gdy wszystkie krawędzie opuszczające wierzchołek v są zbadane, przeszukiwanie wraca do wierzchołka, z którego v został odwiedzony
- W algorytmie definiowany jest poprzednik wierzchołka $u \in V$, który jest zapamiętany w zmiennej $poprz[u]$
- Podgraf poprzedników może składać się z kilku drzew (przeszukiwanie może być wykonywane z wielu źródeł)
- Podgraf poprzedników definiowany jest jako:

$$E_{poprz} = (poprz[v], v) : v \in V \wedge poprz[v] \neq \emptyset$$

- Podgraf poprzedników w przeszukiwaniu wgłąb jest lasem przeszukiwania wgłąb, złożonym z drzew przeszukiwania wgłąb. Krawędzie ze zbioru E_{poprz} to krawędzie drzewowe

Algorytm przeszukiwania grafu wgłęb. Założenia

- W algorytmie wierzchołki są kolorowane na biało, szaro lub czarno
- Kolor wierzchołka zmienia się na szary, gdy jest odwiedzany po raz pierwszy
- Wierzchołek jest kolorowany na czarno, gdy zostaje przetworzony, tzn. jeśli lista jego sąsiedztwa zostaje całkowicie zbadana
- Każdemu wierzchołkowi v przypisuje się dwie etykiety:
 - $d[v]$ – numer kroku obliczeń, w którym v jest odwiedzany po raz pierwszy (gdy wierzchołek kolorowany jest na szaro)
 - $f[v]$ – numer kroku w przeszukiwaniu, w którym kończy się badanie listy sąsiedztwa wierzchołka v (gdy wierzchołek kolorowany jest na czarno)
- Wierzchołek v jest biały do kroku $d[v]$, szary w krokach od $d[v]$ do $f[v]$, a potem czarny

Algorytm przeszukiwania włąb grafów - cd

```
Przeszukaj_wglab(G){
  for(każdy wierzchołek u ∈ V[G]){
    kolor[u] = biały; poprz[u] = -1; }
  czas=0;
  for(każdy wierzchołek u ∈ [G]){
    if(kolor(u)==biały)
      Odwiedz(u);
  }
}

Odwiedz(u){
  kolor[u] = szary; czas=czas+1; d[u] = czas;
  for(każdy wierzchołek v ∈ Ls[u]
    if(kolor[v]==biały){
      poprz[v]=u;
      Odwiedz(v);
    }
  kolor[u] = czarny; czas=czas+1; f[u] = czas;
}
```