Matematyka dyskretna

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. zimowy 2025/26

inf.ug.edu.pl/~amb/MaD

Grafy skierowane

Graf skierowany

- graf skierowany $G = (V, A)$ gdzie $A \subseteq \{(v, w) \in V \times V \mid v \neq w\}$
- tym razem krawędzie (łuki) mają kierunek „od v do w"
  początek i koniec krawędzi
- krawędź skierowana (v, w) to co innego niż (w, v), mogą istnieć obie niezależnie
- podstawowa definicja zakłada, że nie ma pętelek (v, v)
- w razie potrzeby można też rozpatrywać grafy z pętelkami
- można też rozpatrywać grafy z etykietowanymi łukami i/lub wierzchołkami $\ell_v : V \to L_v$, $\ell_a : A \to L_a$
- nośnikiem grafu skierowanego (V, A) jest graf nieskierowany (V, E) gdzie $E = \{\{v, w\} \mid (v, w) \in A\}$
  krawędzie bez zwracania uwagi na zwrot

Ścieżki i cykle

- ścieżką jest ciąg różnych wierzchołków $v_0, \ldots, v_n$ takich, że $(v_i, v_{i+1}) \in A$, $i = 0, \ldots, n-1$
- cyklem jest ciąg $v_0, \ldots, v_n, v_0$ taki, że $v_0, v_1, \ldots, v_n$ jest ścieżką i $(v_n, v_0) \in A$
- cykl może być bardzo krótki, v,w,v jeśli $(v, w) \in A$ i $(w, v) \in A$, zaangażowane są tylko dwa wierzchołki
  (dla grafów nieskierowanych cykl musi zawierać co najmniej trzy wierzchołki)
- graf jest silnie spójny jeśli każde wierzchołki łączy ścieżka
- graf jest acykliczny jeśli nie ma cykli
- graf silnie spójny ma cykle (od v do w i z powrotem)
- w nośniku grafu mogą istnieć ścieżki i cykle nieobecne w grafie

Drzewa jako grafy skierowane

- drzewo jest grafem (nieskierowanym) spójnym i acyklicznym,
- dla każdych dwóch wierzchołków istnieje jednoznaczna ścieżka łącząca te wierzchołki,
- wybranie jednego wierzchołka jako korzenia umożliwia skierowanie wszystkich krawędzi,
  jeśli $v_0 \in V$ jest wybranym korzeniem, to $(v, w) \in A$ jeśli $\{v, w\} \in T$ oraz jedyna ścieżka od v_0 do w ma $\{v, w\}$ jako ostatni krok,
- powstały graf będzie acykliczny
- ale będzie b. daleki od silnej spójności, jeśli będzie istnieć ścieżka od v do w, to znaczy, że v leży po drodze na ścieżce od korzenia v_0 do w, na pewno nie będzie wówczas istnieć ścieżka od w do v.

Stopnie wierzchołków, źródła i zlewy

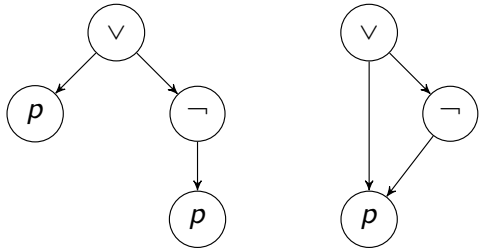
- dwa pojęcia stopnia wierzchołka dla grafów skierowanych,
- $deg^-(v) = |\{w \mid (w, v) \in A\}|$, stopień wejściowy, liczba łuków wchodzących do danego wierzchołka,
  $deg^+(v) = |\{w \mid (v, w) \in A\}|$, stopień wyjściowy, liczba łuków wychodzących z danego wierzchołka,
- źródło (*source*): $deg^-(v) = 0$ i $deg^+(v) > 0$, do wierzchołka nie dochodzą żadne łuki, tylko wychodzą,
  zlew (*sink*): $deg^+(v) = 0$ i $deg^-(v) > 0$, z wierzchołka nie wychodzą żadne łuki, tylko dochodzą,
  wierzchołek izolowany: nie ma żadnych sąsiadów,
  $deg^-(v) = deg^+(v) = 0$,

Grafy acykliczne

- DAG (*directed acyclic graph*)
- Tw.: dla grafów skończonych istnieje co najmniej jedno źródło i co najmniej jeden zlew
- dw.: jeśli wierzchołek nie jest źródłem, możemy się cofnąć o krok
  jest skończona liczba wierzchołków
  na pewno nie wrócimy do już odwiedzonych wierzchołków,
  podobnie dla zlewu
- Tw. można ponumerować wierzchołki zgodnie z kierunkiem łuków $f : V \to \mathbb{N}$ tak by $(v, w) \in A$ implikowało $f(v) < f(w)$
- dw.: źródłom dajemy początkowe numery, po ich usunięciu numerujemy resztę grafu dalszymi numerami

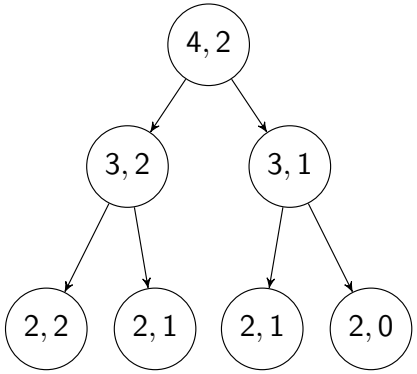
Grafy acykliczne – przykład: wyrażenia arytmetyczne czy logiczne

- np. logika zdaniowa
  - zdanie atomowe jest wyrażeniem
  - $(\neg w)$, $(w_1 \wedge w_2)$, $(w_1 \vee w_2)$ są wyrażeniami
- każde wyrażenie definiuje drzewo rozbioru gramatycznego
  - m.in.: zdanie atomowe jest liściem
- ale czasami jest ważne, że to jest to samo zdanie atomowe powtarzające się w wyrażeniu
- $p \vee \neg p$ ma w korzeniu $\vee$, lewym synem jest p, prawym bardziej skomplikowane wyrażenie, $\neg p$,



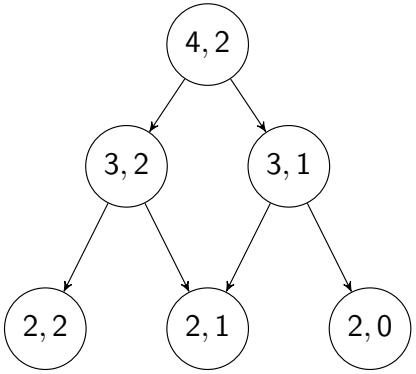
Grafy acykliczne – przykład: opis algorytmu

- np. definicja rekursywna
  w wersji naiwnej
  int newton(int n, int k) if
  $(k == 0)$ return 1;
  else if $(k == n)$ return 1;
  else return newton
  $(n-1,k)$+newton$(n-1,k-1)$;
  sugeruje powtarzające się
  rozgałęzienie na dwa
  podzadania (drzewo binarne)



Grafy acykliczne – przykład: opis algorytmu

- ale w rzeczywistości obliczenia
  będą się powtarzać, DAG lepiej
  oddaje ideę tych obliczeń niż
  drzewo
- strzałka oznacza „aby obliczyć
  v trzeba obliczyć w"
  z tego wniosek: „obliczamy w
  i wynik wykorzystujemy dla v"
- pętla for $(n = 1;;n{+}{+})$
  for $(k = 1;k < n;k{+}{+})$
  newton$[n][k]$=newton$[n-1][k]$+newton$[n-1][k-1]$;
  wykorzystuje zapisane
  wcześniejsze wyniki i likwiduje
  wykładniczą złożoność obliczeń



Najkrótsze ścieżki

- założenie: graf (V, A) ma przypisane wagi (długości) łuków $f : A \to \mathbb{R}$
- czasami żądamy, żeby były dodatnie (albo nieujemne)
- długość ścieżki/cyklu $v_0, \ldots, v_n$ $d = \sum_{i=0}^{n-1} f(v_i, v_{i+1})$
- jeśli istnieje cykl o ujemnej długości, to nie ma najkrótszej ścieżki:
  iterowanie cyklu zmniejsza długość
- zadanie: znaleźć najkrótszą ścieżkę z v do w, o ile istnieje (jeśli nie
  ma żadnej, to odległość $= \infty$)
- Tw: jeśli w grafie nie ma cykli o ujemnej długości, to dla każdej pary
  wierzchołków istnieje najkrótsza ścieżka

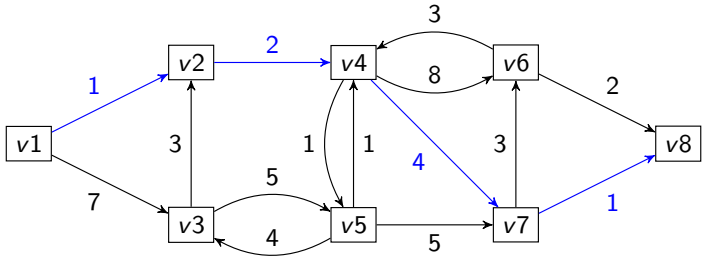
Algorytmy szukania najkrótszej ścieżki

- dany graf $(V, A, f : A \to \mathbb{N})$ i wierzchołki $v, w \in V$
- dwa etapy:
  1. budowa tablicy D najkrótszych dróg z v
  2. znalezienie ścieżki z v do w o odległości $D(w)$
- etap 2: ostatnim łuk w najkrótszej ścieżce $v, \ldots, t, w$ spełnia
  $D(w) = D(t) + f(t, w)$
  po przejrzeniu tablicy D można znaleźć przedostatni wierzchołek
  ścieżki ujawniający skąd wartość $D(w)$
- etap 1: różne algorytmy o różnych złożonościach w zależności od
  kontekstu

Algorytm Dijkstry

- założenie: długości są nieujemne
- tablica D oraz M zbiór wierzchołków nieodwiedzonych
  iteracje algorytmu dopóki M jest niepusty
- $D(v) = 0$, dla innych $w \in V$: $D(w) = f(v, w)$, ∞ jeśli $(v, w) \notin A$, $M = V \setminus \{v\}$
- dopóki $M \neq \varnothing$
  szukamy $t \in M$ taki, że $D(t) = min\{D(w) \mid w \in M\}$
  $M = M \setminus \{t\}$;
  dla $u \in M$: $D(u) = min(D(u), D(t) + f(t, u))$
  czyli może się okazać, że ścieżka od v do u przez t jest krótsza niż dotychczas znana
- złożoność: $O(|V|)^2$ – podwójna pętla po zbiorze wierzchołków

Algorytm Dijkstry – przykład



t	M	D
v_1	$\{v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$	$v_1 \mapsto 0$, $v_2 \mapsto 1$, $v_3 \mapsto 7$, $v_i \mapsto \infty$
v_2	$\{v_3, v_4, v_5, v_6, v_7, v_8\}$	$v_4 \mapsto 3$
v_4	$\{v_3, v_5, v_6, v_7, v_8\}$	$v_5 \mapsto 4$, $v_6 \mapsto 11$, $v_7 \mapsto 7$
v_5	$\{v_3, v_6, v_7, v_8\}$	v_3, v_5 bez zmian
v_7	$\{v_3, v_6, v_8\}$	$v_6 \mapsto 10$, $v_8 \mapsto 8$
v_3	$\{v_6, v_8\}$	v_2, v_5 bez zmian
v_8	$\{v_6\}$	

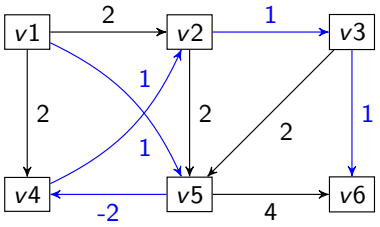
Algorytm dla grafu acyklicznego

- dany graf $(V, A, f : A \to \mathbb{N})$ z numeracją wierzchołków $g : V \to \mathbb{N}$ zgodną kierunkiem łuków
- budujemy tablicę D odległości od ustalonego v, $D(v) = 0$ pozostałe ∞
- dla każdego wierzchołka u po kolei wg numeracji g przeglądamy $\{t \in V \mid g(t) < g(u)\}$ w kolejności numeracji
  $D(u) = min(D(u), D(t) + f(t, u))$
- każda ścieżka od v do u przebiega w kolejności numeracji więc najkrótsza musi być znaleziona w tym algorytmie
- złożoność: $O(|V|)^2$ – podwójna pętla po zbiorze wierzchołków

Algorytm Forda Bellmana

- założenie: w grafie nie ma cykli o ujemnej długości, mogą być cykle, mogą być ujemne długości poszczególnych łuków
- tablica D, na początku $D(v) = 0$, dla innych $w \in V$ $D(w) = f(v, w)$ (być może ∞)
- dla wszystkich u, dla wszystkich t: $D(u) = min(D(u), D(t) + f(t, u))$
- powyższa podwójna pętla jest powtarzana $|V|$ krotnie
- po k iteracji tablica D zawiera najkrótsze ścieżki długości k, najkrótsza ścieżka ma długość $< |V|$ więc algorytm poprawnie ją znajdzie
- złożoność: $O(|V|)^3$ – potrójna pętla długości $|V|$ każda

Algorytm Forda Bellmana – przykład



nr	*D*
1	$v_1 \mapsto 0$, $v_2 \mapsto 2$, $v_3 \mapsto \infty$, $v_4 \mapsto 2$, $v_5 \mapsto 1$, $v_6 \mapsto \infty$
2	$v_1 \mapsto 0$, $v_2 \mapsto 2$, $v_3 \mapsto 3$, $v_4 \mapsto -1$, $v_5 \mapsto 1$, $v_6 \mapsto 4$
3	$v_1 \mapsto 0$, $v_2 \mapsto 0$, $v_3 \mapsto 1$, $v_4 \mapsto -1$, $v_5 \mapsto 1$, $v_6 \mapsto 2$
4	bez zmian