Złożoność obliczeniowa - Transformacja wielomianowa - MATCHING - MAX FLOW - REACHABILITY

Marcin Żurowski

01 kwietnia 2026

Plan zajęć

Osiągalność w grafie (REACHABILITY)

Dany jest graf $G = (V, E)$ oraz dwa jego wierzchołki $v_1, v_2 \in V$.
Czy istnieje w grafie ścieżka od v_1 do v_2.

Algorytm rozwiązujący problem REACHABILITY

- Tworzymy zbiór wierzchołków $S = \{1\}$.
- Tworzymy tablicę przejścia p i wypełniamy ją wartościami -1
- Tworzymy tablicę Z w której zaznaczamy wierzchołek, który był w zbiorze S.
- Dopóki $S \neq \phi$ usuwamy z niego dowolny wierzchołek i i dodajemy wszystkie wierzchołki j dla których istnieje krawędź (i, j). Zaznaczamy w Z wszystkie dodane wierzchołki (dla grafu ważonego wpisujemy wartość $\min\{Z[j], c(i, j)\}$).
- w tablicy p pod wszystkimi indeksami j wpisujemy wartości i.
- Jeżeli wierzchołek n jest zaznaczony to istnieje ścieżka od 1 do n w grafie G.

Algorytm rozwiązujący problem REACHABILITY

- Tworzymy zbiór wierzchołków $S = \{1\}$.
- Tworzymy tablicę przejścia p i wypełniamy ją wartościami -1
- Tworzymy tablicę Z w której zaznaczamy wierzchołek, który był w zbiorze S.
- Dopóki $S \neq \phi$ usuwamy z niego dowolny wierzchołek i i dodajemy wszystkie wierzchołki j dla których istnieje krawędź (i, j). Zaznaczamy w Z wszystkie dodane wierzchołki (dla grafu ważonego wpisujemy wartość $\min\{Z[j], c(i, j)\}$).
- w tablicy p pod wszystkimi indeksami j wpisujemy wartości i.
- Jeżeli wierzchołek n jest zaznaczony to istnieje ścieżka od 1 do n w grafie G.

Algorytm rozwiązujący problem REACHABILITY

- Tworzymy zbiór wierzchołków $S = \{1\}$.
- Tworzymy tablicę przejścia p i wypełniamy ją wartościami -1
- Tworzymy tablicę Z w której zaznaczamy wierzchołek, który był w zbiorze S.
- Dopóki $S \neq \phi$ usuwamy z niego dowolny wierzchołek i i dodajemy wszystkie wierzchołki j dla których istnieje krawędź (i, j). Zaznaczamy w Z wszystkie dodane wierzchołki (dla grafu ważonego wpisujemy wartość $\min\{Z[j], c(i, j)\}$).
- w tablicy p pod wszystkimi indeksami j wpisujemy wartości i.
- Jeżeli wierzchołek n jest zaznaczony to istnieje ścieżka od 1 do n w grafie G.

Algorytm rozwiązujący problem REACHABILITY

- Tworzymy zbiór wierzchołków $S = \{1\}$.
- Tworzymy tablicę przejścia p i wypełniamy ją wartościami -1
- Tworzymy tablicę Z w której zaznaczamy wierzchołek, który był w zbiorze S.
- Dopóki $S \neq \phi$ usuwamy z niego dowolny wierzchołek i i dodajemy wszystkie wierzchołki j dla których istnieje krawędź (i, j). Zaznaczamy w Z wszystkie dodane wierzchołki (dla grafu ważonego wpisujemy wartość $\min\{Z[j], c(i, j)\}$).
- w tablicy p pod wszystkimi indeksami j wpisujemy wartości i.
- Jeżeli wierzchołek n jest zaznaczony to istnieje ścieżka od 1 do n w grafie G.

Algorytm rozwiązujący problem REACHABILITY

- Tworzymy zbiór wierzchołków $S = \{1\}$.
- Tworzymy tablicę przejścia p i wypełniamy ją wartościami -1
- Tworzymy tablicę Z w której zaznaczamy wierzchołek, który był w zbiorze S.
- Dopóki $S \neq \phi$ usuwamy z niego dowolny wierzchołek i i dodajemy wszystkie wierzchołki j dla których istnieje krawędź (i, j). Zaznaczamy w Z wszystkie dodane wierzchołki (dla grafu ważonego wpisujemy wartość $\min\{Z[j], c(i, j)\}$).
- w tablicy p pod wszystkimi indeksami j wpisujemy wartości i.
- Jeżeli wierzchołek n jest zaznaczony to istnieje ścieżka od 1 do n w grafie G.

Algorytm rozwiązujący problem REACHABILITY

- Tworzymy zbiór wierzchołków $S = \{1\}$.
- Tworzymy tablicę przejścia p i wypełniamy ją wartościami -1
- Tworzymy tablicę Z w której zaznaczamy wierzchołek, który był w zbiorze S.
- Dopóki $S \neq \phi$ usuwamy z niego dowolny wierzchołek i i dodajemy wszystkie wierzchołki j dla których istnieje krawędź (i, j). Zaznaczamy w Z wszystkie dodane wierzchołki (dla grafu ważonego wpisujemy wartość $\min\{Z[j], c(i, j)\}$).
- w tablicy p pod wszystkimi indeksami j wpisujemy wartości i.
- Jeżeli wierzchołek n jest zaznaczony to istnieje ścieżka od 1 do n w grafie G.

Maksymalny przepływ (MAX FLOW)

Dana jest sieć $N = (V, E, s, t, c)$, gdzie $s, t \in V$, natomiast c jest funkcją, która dla krawędzi (i, j) przyporządkowuje jej wagę będącą liczbą naturalną. Musimy znaleźć maksymalny przepływ, czyli funkcję f, która każdej krawędzi (i, j) przyporządkuje wartość taką, że $f(i, j) \leq c(i, j)$, dla każdego węzła z wyjątkiem s, t suma wartości funkcja f krawędzi wchodzących jest równa sumie wartości funkcji f dla krawędzi wychodzących.

Algorytm rozwiązujący problem MAX FLOW

- Tworzymy dwa grafy (macierze sąsiedztwa): jeden pusty (macierz wypełniona zerami) reprezentujący przepływ f, drugi $N(f) = N$ (macierz wypełniona wagami krawędzi sieci N).
- Znajdujemy ścieżkę w grafie $N(f)$ od s do t.
- Jeśli nie znaleźliśmy ścieżki w $N(f)$, to mamy maksymalny przepływ.
- Jeśli znaleźliśmy ścieżkę p w grafie to:

Algorytm rozwiązujący problem MAX FLOW

- Tworzymy dwa grafy (macierze sąsiedztwa): jeden pusty (macierz wypełniona zerami) reprezentujący przepływ f, drugi $N(f) = N$ (macierz wypełniona wagami krawędzi sieci N).
- Znajdujemy ścieżkę w grafie $N(f)$ od s do t.
- Jeśli nie znaleźliśmy ścieżki w $N(f)$, to mamy maksymalny przepływ.
- Jeśli znaleźliśmy ścieżkę p w grafie to:
  - do f dodajemy ścieżkę p o najmniejszej wadze z $N(f)$ tworzymy $N(f) = (V, E, s, t, c)$ nowe wagi

Algorytm rozwiązujący problem MAX FLOW

- Tworzymy dwa grafy (macierze sąsiedztwa): jeden pusty (macierz wypełniona zerami) reprezentujący przepływ f, drugi $N(f) = N$ (macierz wypełniona wagami krawędzi sieci N).
- Znajdujemy ścieżkę w grafie $N(f)$ od s do t.
- Jeśli nie znaleźliśmy ścieżki w $N(f)$, to mamy maksymalny przepływ.
- Jeśli znaleźliśmy ścieżkę p w grafie to:
  - dodajemy do f ścieżkę p o najmniejszej wadze
  - z $N(f)$ tworzymy $N'(f) = (V, E', s, t, c')$ następująco:

Algorytm rozwiązujący problem MAX FLOW

- Tworzymy dwa grafy (macierze sąsiedztwa): jeden pusty (macierz wypełniona zerami) reprezentujący przepływ f, drugi $N(f) = N$ (macierz wypełniona wagami krawędzi sieci N).
- Znajdujemy ścieżkę w grafie $N(f)$ od s do t.
- Jeśli nie znaleźliśmy ścieżki w $N(f)$, to mamy maksymalny przepływ.
- Jeśli znaleźliśmy ścieżkę p w grafie to:
  - dodajemy do f ścieżkę p o najmniejszej wadze
  - z $N(f)$ tworzymy $N'(f) = (V, E', s, t, c')$ następująco:
    - $E' = \{(u,v) : c(u,v) - c(u,p) + c(p,v) > 0 \text{ lub } c(v,u) + c(u,p) - c(p,v) > 0\}$
    - $c'(u,v) = c(u,v) - c(u,p) + c(p,v)$
    - $c'(v,u) = c(v,u) + c(u,p) - c(p,v)$

Algorytm rozwiązujący problem MAX FLOW

- Tworzymy dwa grafy (macierze sąsiedztwa): jeden pusty (macierz wypełniona zerami) reprezentujący przepływ f, drugi $N(f) = N$ (macierz wypełniona wagami krawędzi sieci N).
- Znajdujemy ścieżkę w grafie $N(f)$ od s do t.
- Jeśli nie znaleźliśmy ścieżki w $N(f)$, to mamy maksymalny przepływ.
- Jeśli znaleźliśmy ścieżkę p w grafie to:
  - dodajemy do f ścieżkę p o najmniejszej wadze
  - z $N(f)$ tworzymy $N'(f) = (V, E', s, t, c')$ następująco:
    - $E' = E - \{(i,j) : f(i,j) = c(i,j)\} \cup \{(i,j) : (j,i) \in E, f(j,i) > 0\}$
    - $c'(i,j) = c(i,j) - f(i,j)$ dla $(i,j) \in E$
      $c'(i,j) = f(j,i)$ dla $(i,j) \in E' - E$

Algorytm rozwiązujący problem MAX FLOW

- Tworzymy dwa grafy (macierze sąsiedztwa): jeden pusty (macierz wypełniona zerami) reprezentujący przepływ f, drugi $N(f) = N$ (macierz wypełniona wagami krawędzi sieci N).
- Znajdujemy ścieżkę w grafie $N(f)$ od s do t.
- Jeśli nie znaleźliśmy ścieżki w $N(f)$, to mamy maksymalny przepływ.
- Jeśli znaleźliśmy ścieżkę p w grafie to:
  - dodajemy do f ścieżkę p o najmniejszej wadzę
  - z $N(f)$ tworzymy $N'(f) = (V, E', s, t, c')$ następująco:
    - $E' = E - \{(i,j) : f(i,j) = c(i,j)\} \cup \{(i,j) : (j,i) \in E, f(j,i) > 0\}$
    - $c'(i,j) = c(i,j) - f(i,j)$ dla $(i,j) \in E$
      $c'(i,j) = f(j,i)$ dla $(i,j) \in E' - E$

Algorytm rozwiązujący problem MAX FLOW

- Tworzymy dwa grafy (macierze sąsiedztwa): jeden pusty (macierz wypełniona zerami) reprezentujący przepływ f, drugi $N(f) = N$ (macierz wypełniona wagami krawędzi sieci N).
- Znajdujemy ścieżkę w grafie $N(f)$ od s do t.
- Jeśli nie znaleźliśmy ścieżki w $N(f)$, to mamy maksymalny przepływ.
- Jeśli znaleźliśmy ścieżkę p w grafie to:
  - dodajemy do f ścieżkę p o najmniejszej wadze
  - z $N(f)$ tworzymy $N'(f) = (V, E', s, t, c')$ następująco:
    - $E' = E - \{(i,j) : f(i,j) = c(i,j)\} \cup \{(i,j) : (j,i) \in E, f(j,i) > 0\}$
    - $c'(i,j) = c(i,j) - f(i,j)$ dla $(i,j) \in E$
      $c'(i,j) = f(j,i)$ dla $(i,j) \in E' - E$

Algorytm rozwiązujący problem MAX FLOW

- Tworzymy dwa grafy (macierze sąsiedztwa): jeden pusty (macierz wypełniona zerami) reprezentujący przepływ f, drugi $N(f) = N$ (macierz wypełniona wagami krawędzi sieci N).
- Znajdujemy ścieżkę w grafie $N(f)$ od s do t.
- Jeśli nie znaleźliśmy ścieżki w $N(f)$, to mamy maksymalny przepływ.
- Jeśli znaleźliśmy ścieżkę p w grafie to:
  - dodajemy do f ścieżkę p o najmniejszej wadze
  - z $N(f)$ tworzymy $N'(f) = (V, E', s, t, c')$ następująco:
    - $E' = E - \{(i,j) : f(i,j) = c(i,j)\} \cup \{(i,j) : (j,i) \in E, f(j,i) > 0\}$
    - $c'(i,j) = c(i,j) - f(i,j)$ dla $(i,j) \in E$
      $c'(i,j) = f(j,i)$ dla $(i,j) \in E' - E$

Skojarzenie dwudzielne(MATCHING)

Dany jest graf dwudzielny $B = (U, V, E)$ oraz $E \subseteq U \times V$. Znaleźć maksymalny zbiór $M \subseteq E$, taki że każda krawędź z M nie jest incydentna z inną krawędzią ze zbioru M.

Algorytm rozwiązujący problem MATCHING

- Tworzymy sieć $N = (V', E', s, t, c)$ w następujący sposób:
  - $V' = U \cup V \cup \{s, t\}$
  - $E' = \{(s, u) : u \in U\} \cup E \cup \{(v, t) : v \in V\}$
  - s, t - dwa nowe wierzchołki
  - $c(i, j) = 1$ dla wszystkich krawędzi E'
- Znajdujemy MAX FLOW w N.

Algorytm rozwiązujący problem MATCHING

- Tworzymy sieć $N = (V', E', s, t, c)$ w następujący sposób:
  - $V' = U \cup V \cup \{s, t\}$
  - $E' = \{(s, u) : u \in U\} \cup E \cup \{(v, t) : v \in V\}$
  - s, t - dwa nowe wierzchołki
  - $c(i, j) = 1$ dla wszystkich krawędzi E'
- Znajdujemy MAX FLOW w N.

Algorytm rozwiązujący problem MATCHING

- Tworzymy sieć $N = (V', E', s, t, c)$ w następujący sposób:
  - $V' = U \cup V \cup \{s, t\}$
  - $E' = \{(s, u) : u \in U\} \cup E \cup \{(v, t) : v \in V\}$
  - s, t - dwa nowe wierzchołki
  - $c(i, j) = 1$ dla wszystkich krawędzi E'
- Znajdujemy MAX FLOW w N.

Algorytm rozwiązujący problem MATCHING

- Tworzymy sieć $N = (V', E', s, t, c)$ w następujący sposób:
  - $V' = U \cup V \cup \{s, t\}$
  - $E' = \{(s, u) : u \in U\} \cup E \cup \{(v, t) : v \in V\}$
  - s, t - dwa nowe wierzchołki
  - $c(i, j) = 1$ dla wszystkich krawędzi E'
- Znajdujemy MAX FLOW w N.

Algorytm rozwiązujący problem MATCHING

- Tworzymy sieć $N = (V', E', s, t, c)$ w następujący sposób:
  - $V' = U \cup V \cup \{s, t\}$
  - $E' = \{(s, u) : u \in U\} \cup E \cup \{(v, t) : v \in V\}$
  - s, t - dwa nowe wierzchołki
  - $c(i, j) = 1$ dla wszystkich krawędzi E'
- Znajdujemy MAX FLOW w N.

Algorytm rozwiązujący problem MATCHING

- Tworzymy sieć $N = (V', E', s, t, c)$ w następujący sposób:
  - $V' = U \cup V \cup \{s, t\}$
  - $E' = \{(s, u) : u \in U\} \cup E \cup \{(v, t) : v \in V\}$
  - s, t - dwa nowe wierzchołki
  - $c(i, j) = 1$ dla wszystkich krawędzi E'
- Znajdujemy MAX FLOW w N.

Praca domowa 5 pkt.

Napisz program, który rozwiązuje problemy:

- REACHABILITY
- MAX FLOW przekształcając go w problem REACHABILITY
- MATCHING przekształcając go w problem MAX FLOW

Praca domowa 5 pkt.

Napisz program, który rozwiązuje problemy:

- REACHABILITY
- MAX FLOW przekształcając go w problem REACHABILITY
- MATCHING przekształcając go w problem MAX FLOW

Praca domowa 5 pkt.

Napisz program, który rozwiązuje problemy:

- REACHABILITY
- MAX FLOW przekształcając go w problem REACHABILITY
- MATCHING przekształcając go w problem MAX FLOW