Contracting a Planar Graph Efficiently

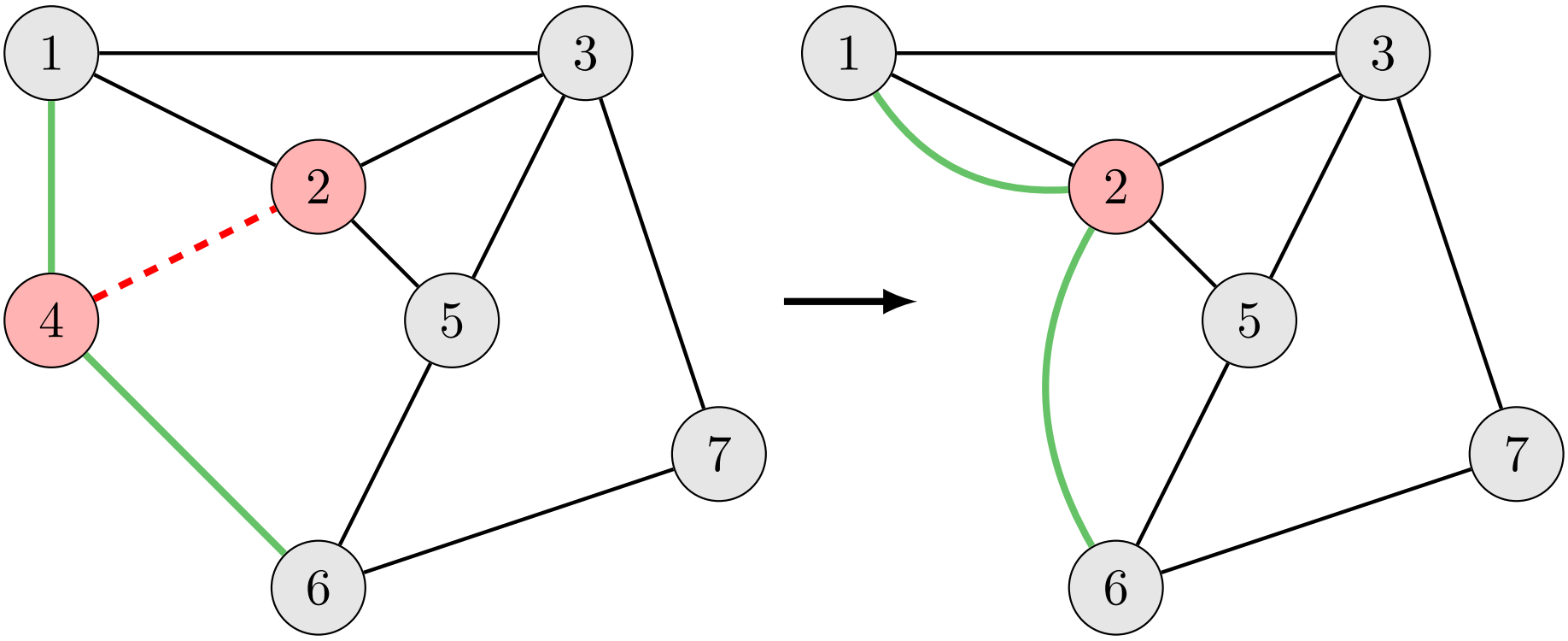
Autorzy: Jacob Holm, Giuseppe F. Italiano, Adam Karczmarz, Jakub Łącki, Eva Rotenberg, Piotr Sankowski

Autor prezentacji: Jakub Łabaj, 15 listopada 2018

Plan prezentacji

1. Co to jest kontrakcja
2. Kontrakcja w grafach planarnych
3. Osiągnięcia pracy
4. r-podział
5. Algorytm
6. Struktura scalająca wierzchołki
7. Wielopoziomowa struktura

Kontrakcja krawędzi grafu



Kontrakcja w grafach planarnych

- Graf nieskierowany jest **planarny** wtw. kiedy nie zawiera jako minor kliki K_5 ani grafu $K_{3,3}$.

Kontrakcja w grafach planarnych

- Graf nieskierowany jest **planarny** wtw. kiedy nie zawiera jako minor kliki K_5 ani grafu $K_{3,3}$.
- **Minor** to podgraf osiągany za pomocą za pomocą **kontrakcji**, usuwania krawędzi i usuwania wierzchołków izolowanych.

Kontrakcja w grafach planarnych

- Graf nieskierowany jest **planarny** wtw. kiedy nie zawiera jako minor kliki K_5 ani grafu $K_{3,3}$.
- **Minor** to podgraf osiągany za pomocą za pomocą **kontrakcji**, usuwania krawędzi i usuwania wierzchołków izolowanych.
- Operacja kontrakcji zachowuje planarność.

Kontrakcja w grafach planarnych

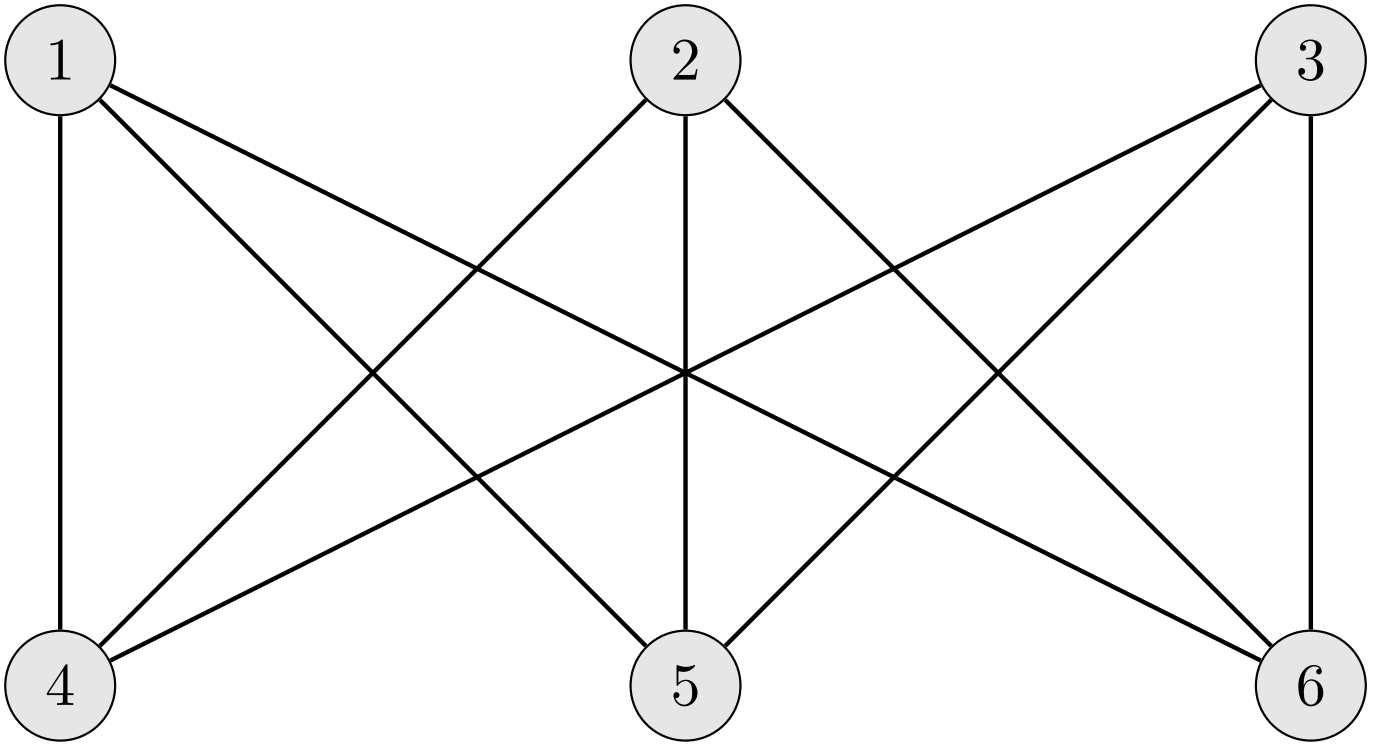
- Graf nieskierowany jest **planarny** wtw. kiedy nie zawiera jako minor kliki K_5 ani grafu $K_{3,3}$.
- **Minor** to podgraf osiągany za pomocą za pomocą **kontrakcji**, usuwania krawędzi i usuwania wierzchołków izolowanych.
- Operacja kontrakcji zachowuje planarność.
- Algorytmy bazujące na kontrakcji, np. 5-kolorowanie, MST.

Kontrakcja w grafach planarnych

- Graf nieskierowany jest **planarny** wtw. kiedy nie zawiera jako minor kliki K_5 ani grafu $K_{3,3}$.
- **Minor** to podgraf osiągany za pomocą za pomocą **kontrakcji**, usuwania krawędzi i usuwania wierzchołków izolowanych.
- Operacja kontrakcji zachowuje planarność.
- Algorytmy bazujące na kontrakcji, np. 5-kolorowanie, MST.

Algorytmy w praktyce: ładny i nieskomplikowany kod, ale nieefektywne - dotychczas brak wydajnej struktury dla kontrakcji.

Graf $K_{3,3}$



Osiągnięcia pracy

Struktura danych dla operacji kontrakcji dla multigrafu planarnego:

Osiągnięcia pracy

Struktura danych dla operacji kontrakcji dla multigrafu planarnego:

- Ciąg operacji kontrakcji w złożoności $O(n)$.

Osiągnięcia pracy

Struktura danych dla operacji kontrakcji dla multigrafu planarnego:

- Ciąg operacji kontrakcji w złożoności $O(n)$.
- Wykrywanie krawędzi równoległych i pętli.

Osiągnięcia pracy

Struktura danych dla operacji kontrakcji dla multigrafu planarnego:

- Ciąg operacji kontrakcji w złożoności $O(n)$.
- Wykrywanie krawędzi równoległych i pętli.
- Odpowiedź na zapytania *"czy x jest połączony krawędzią z y"* w czasie stałym.

Osiągnięcia pracy

Struktura danych dla operacji kontrakcji dla multigrafu planarnego:

- Ciąg operacji kontrakcji w złożoności $O(n)$.
- Wykrywanie krawędzi równoległych i pętli.
- Odpowiedź na zapytania *"czy x jest połączony krawędzią z y"* w czasie stałym.
- Przechowywanie listy sąsiadów i stopnia wierzchołka.

Osiągnięcia pracy

Struktura danych dla operacji kontrakcji dla multigrafu planarnego:

- Ciąg operacji kontrakcji w złożoności $O(n)$.
- Wykrywanie krawędzi równoległych i pętli.
- Odpowiedź na zapytania *"czy x jest połączony krawędzią z y"* w czasie stałym.
- Przechowywanie listy sąsiadów i stopnia wierzchołka.

Najlepiej dotychczas: odpowiedź na zapytania w czasie stałym, ale kontrakcja w czasie $O(log\ n)$.

Osiągnięcia pracy

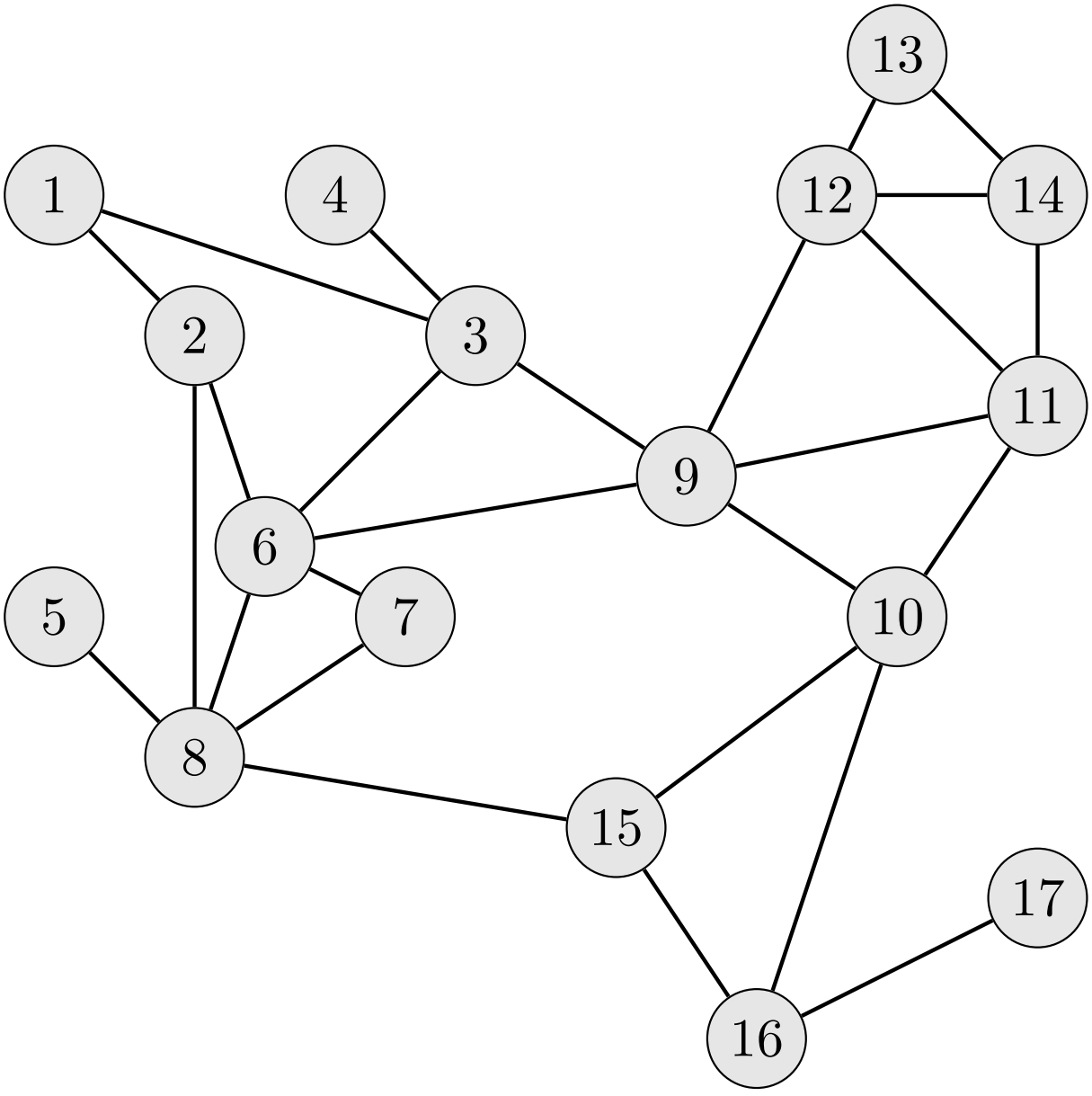
Rezultat: użycie powyższej struktury daje możliwość ładnej i efektywnej implementacji wielu znanych algorytmów.

Osiągnięcia pracy

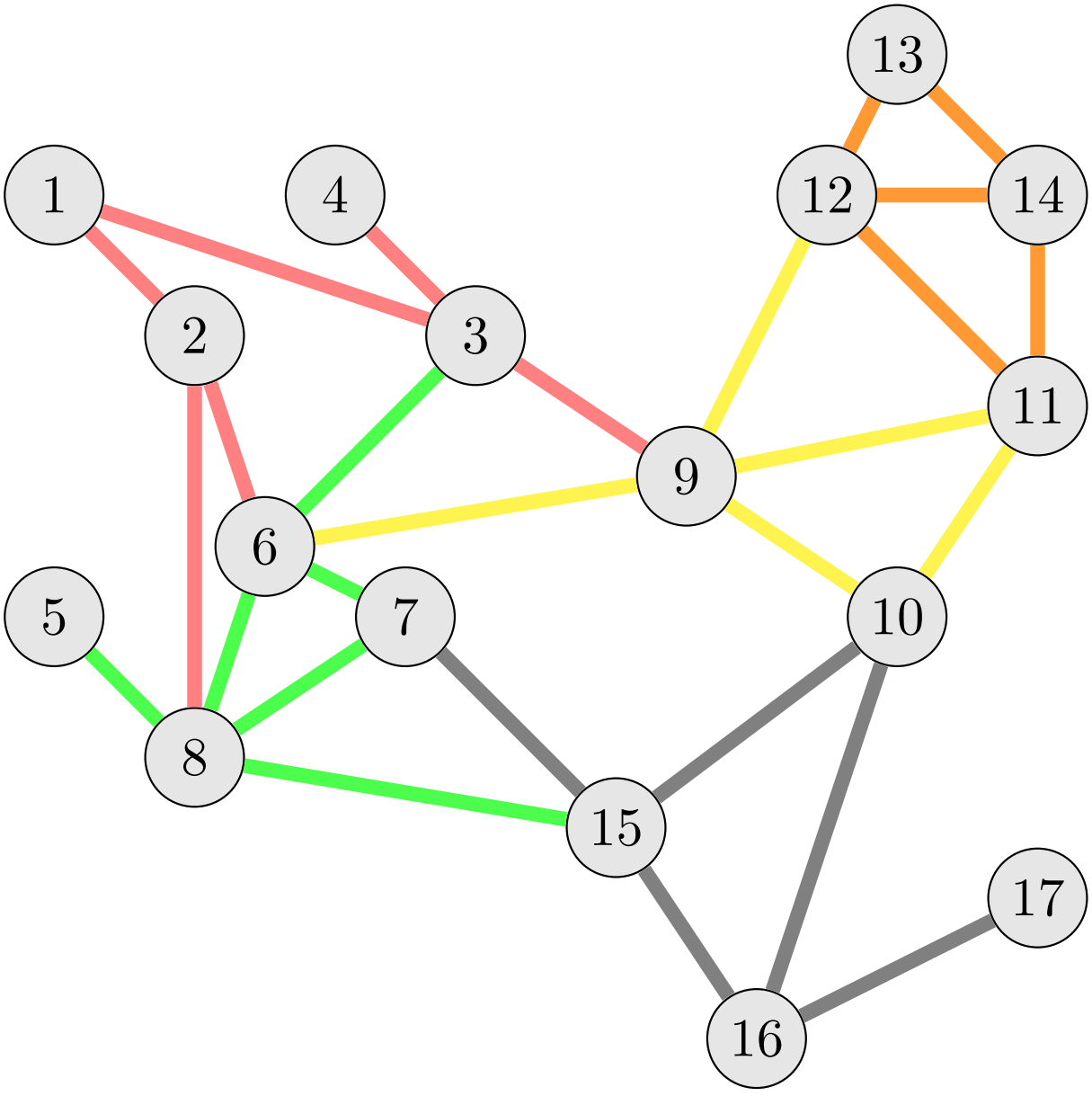
Rezultat: użycie powyższej struktury daje możliwość ładnej i efektywnej implementacji wielu znanych algorytmów.

Ponadto: poprawienie złożoności algorytmów dla niektórych znanych problemów.

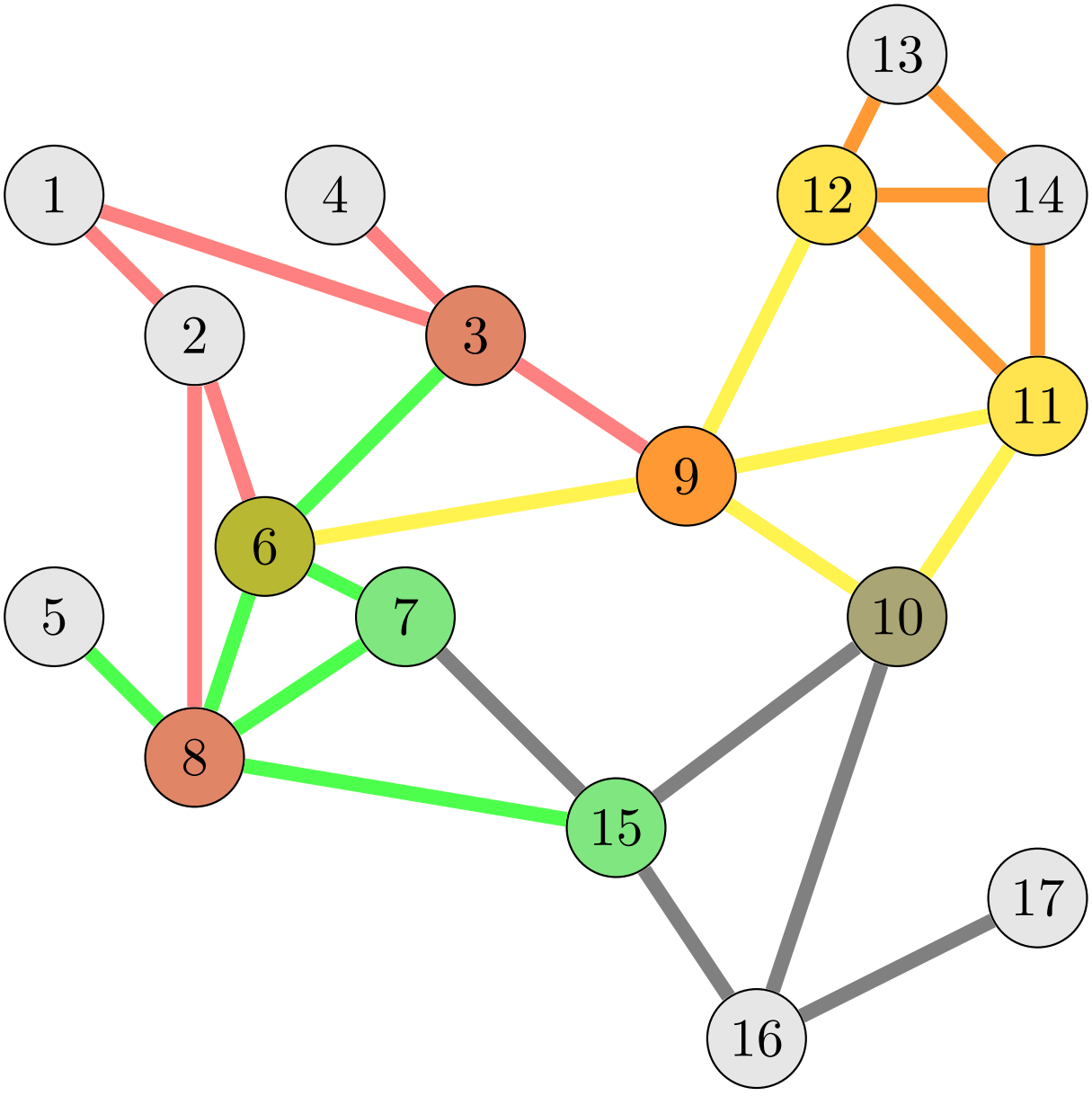
r-podział - graf wejściowy



r-podział - rozłączny krawędziowo



r-podział - wierzchołki graniczne



r-podział formalnie

- Podział grafu na spójne podgrupy
- $O(r)$ wierzchołków w grupie
- $O(n/r)$ rozłącznych grup
- $O(\sqrt{r})$ wierzchołków granicznych w każdej grupie

Fakt: r-podział da się wykonać w czasie liniowym.

Pomysł na algorytm

- Prosta struktura bazująca hashmapie na przeprowadzanie kontrakcji: $O(log\ n)$ (amortyzowany) aktualizacja i $O(1)$ zapytanie

Pomysł na algorytm

- Prosta struktura bazująca hashmapie na przeprowadzanie kontrakcji: $O(log\ n)$ (amortyzowany) aktualizacja i $O(1)$ zapytanie
- r-podział grafu przy $r = log^4\ n$, wtedy wierzchołków granicznych $O(n/log^2\ n)$

Pomysł na algorytm

- Prosta struktura bazująca hashmapie na przeprowadzanie kontrakcji: $O(log\ n)$ (amortyzowany) aktualizacja i $O(1)$ zapytanie
- r-podział grafu przy $r = log^4\ n$, wtedy wierzchołków granicznych $O(n/log^2\ n)$
- Rozpatrujemy zdarzenia osobno dla “granic” i wnętrza

Pomysł na algorytm

- Prosta struktura bazująca hashmapie na przeprowadzanie kontrakcji: $O(log\ n)$ (amortyzowany) aktualizacja i $O(1)$ zapytanie
- r-podział grafu przy $r = log^4\ n$, wtedy wierzchołków granicznych $O(n/log^2\ n)$
- Rozpatrujemy zdarzenia osobno dla "granic" i wnętrza
- Kosztowne operacje: krawędzie przepływające pomiędzy granicami i wnętrzem

Pomysł na algorytm

- Prosta struktura bazująca hashmapie na przeprowadzanie kontrakcji: $O(log\ n)$ (amortyzowany) aktualizacja i $O(1)$ zapytanie
- r-podział grafu przy $r = log^4\ n$, wtedy wierzchołków granicznych $O(n/log^2\ n)$
- Rozpatrujemy zdarzenia osobno dla "granic" i wnętrza
- Kosztowne operacje: krawędzie przepływające pomiędzy granicami i wnętrzem
- Da się osiągnąć złożoność $O(n\ log\ log\ n)$

Pomysł na algorytm - ulepszenie

Zagnieżdżony r-podział:

Pomysł na algorytm - ulepszenie

Zagnieżdżony r-podział:

- Podpodział na podgrafy wielkości $O(log^4\ log^4\ n) = O(log^4\ log\ n)$

Pomysł na algorytm - ulepszenie

Zagnieżdżony r-podział:

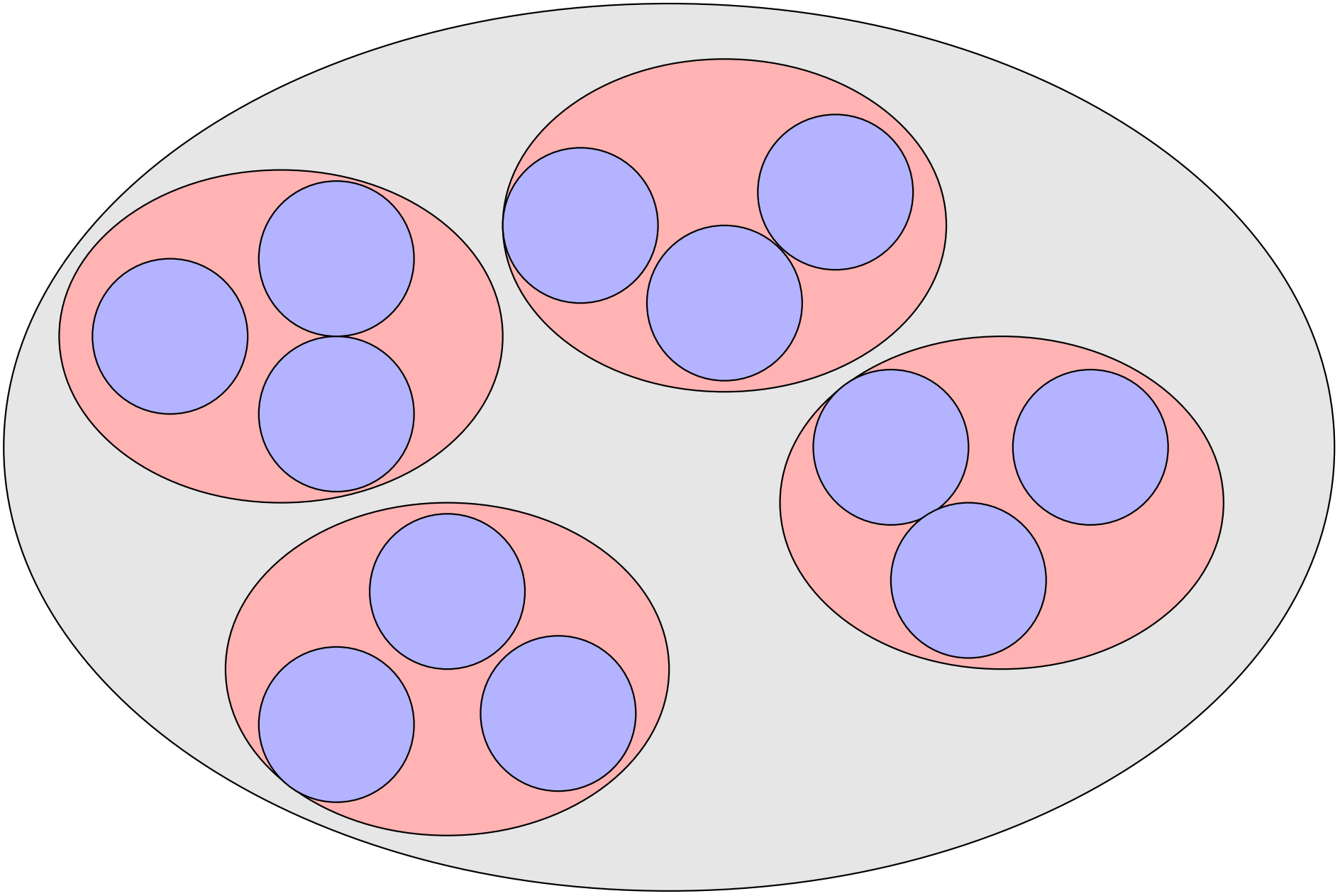
- Podpodział na podgrafy wielkości $O(log^4\ log^4\ n) = O(log^4\ log\ n)$
- Na tyle mało wierzchołków, że możemy obliczyć w $O(n)$ wszystkie możliwe sekwencje kontrakcji

Pomysł na algorytm - ulepszenie

Zagnieżdżony r-podział:

- Podpodział na podgrafy wielkości $O(log^4\ log^4\ n) = O(log^4\ log\ n)$
- Na tyle mało wierzchołków, że możemy obliczyć w $O(n)$ wszystkie możliwe sekwencje kontrakcji
- Złożoność: $O(n)$

Podziały grafu w algorytmie



Struktura scalająca wierzchołki

Struktura działająca w czasie $O(1)$ - zapytanie i $O(log\ n)$ - aktualizacja:

Struktura scalająca wierzchołki

Struktura działająca w czasie $O(1)$ - zapytanie i $O(log\ n)$ - aktualizacja:

- Utrzymywane: graf prosty $G = (V, E)$ oraz zbiór graniczny $B \subset V$

Struktura scalająca wierzchołki

Struktura działająca w czasie $O(1)$ - zapytanie i $O(log\ n)$ - aktualizacja:

- Utrzymywane: graf prosty $G = (V, E)$ oraz zbiór graniczny $B \subset V$
- Operacje: łączenie wierzchołków i dokładanie krawędzi, zachowujące planarność

Struktura scalająca wierzchołki

Struktura działająca w czasie $O(1)$ - zapytanie i $O(log\ n)$ - aktualizacja:

- Utrzymywane: graf prosty $G = (V, E)$ oraz zbiór graniczny $B \subset V$
- Operacje: łączenie wierzchołków i dokładanie krawędzi, zachowujące planarność
- Po każdej operacji zgłaszane krawędzie równoległe (po operacji łączone w jedną) i pętle (usuwane)

Struktura scalająca wierzchołki

Struktura działająca w czasie $O(1)$ - zapytanie i $O(log\ n)$ - aktualizacja:

- Utrzymywane: graf prosty $G = (V, E)$ oraz zbiór graniczny $B \subset V$
- Operacje: łączenie wierzchołków i dokładanie krawędzi, zachowujące planarność
- Po każdej operacji zgłaszane krawędzie równoległe (po operacji łączone w jedną) i pętle (usuwane)
- Zgłaszane i usuwane krawędzie z obydwoma wierzchołkami w B

Struktura scalająca wierzchołki

Rozszerzenie struktury:

Struktura scalająca wierzchołki

Rozszerzenie struktury:

- Dla każdego wierzchołka utrzymywana lista sąsiadów

Struktura scalająca wierzchołki

Rozszerzenie struktury:

- Dla każdego wierzchołka utrzymywana lista sąsiadów
- Funkcja $\alpha(x, y) = e$ - dla dwóch wierzchołków zwraca jedną z krawędzi je łączącą

Struktura scalająca wierzchołki

Rozszerzenie struktury:

- Dla każdego wierzchołka utrzymywana lista sąsiadów
- Funkcja $\alpha(x, y) = e$ - dla dwóch wierzchołków zwraca jedną z krawędzi je łączącą
- $\phi(u) = x$ - dla wierzchołka z oryginalnego grafu zwraca aktualny wierzchołek, do którego został dołączony w trakcie wykonywania operacji

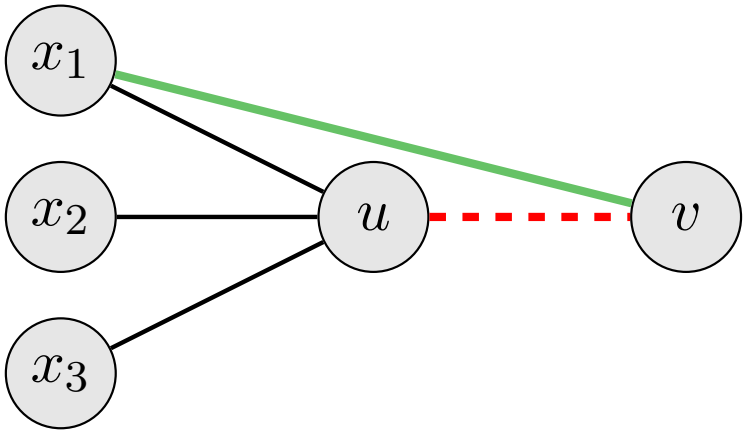
Struktura scalająca wierzchołki

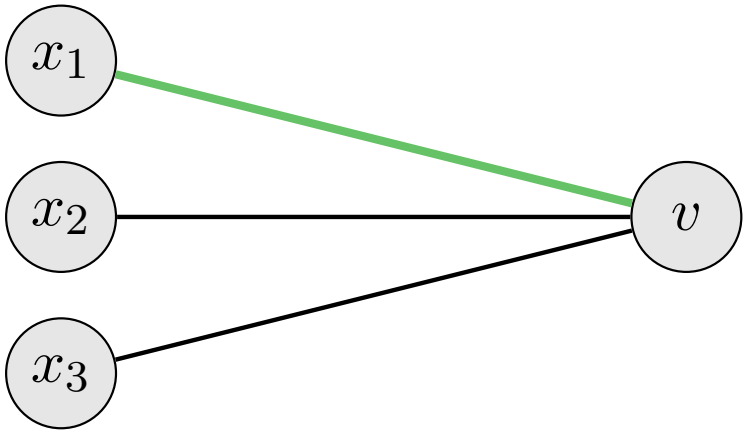
Rozszerzenie struktury:

- Dla każdego wierzchołka utrzymywana lista sąsiadów
- Funkcja $\alpha(x, y) = e$ - dla dwóch wierzchołków zwraca jedną z krawędzi je łączącą
- $\phi(u) = x$ - dla wierzchołka z oryginalnego grafu zwraca aktualny wierzchołek, do którego został dołączony w trakcie wykonywania operacji
- Odwrotność $\phi^{-1}(x)$

Łączenie

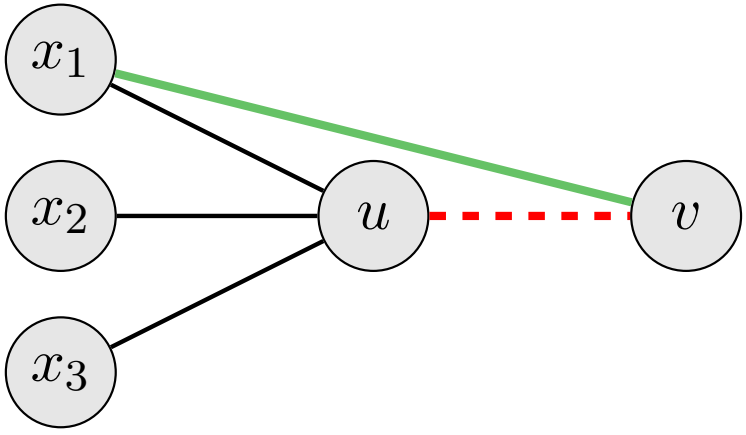
Zwykłe wierzchołki

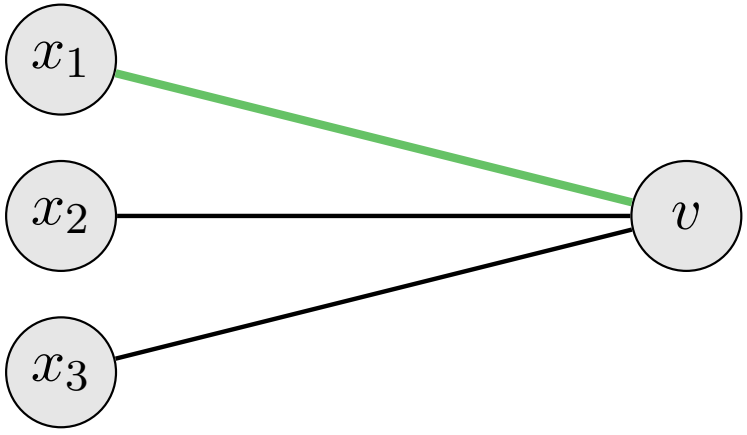




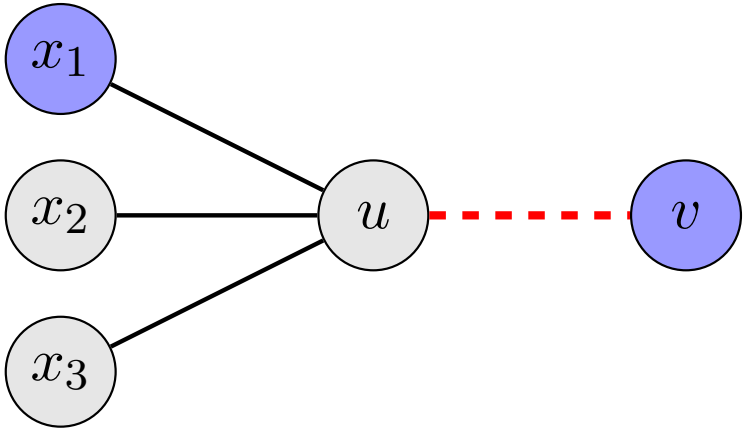
Łączenie

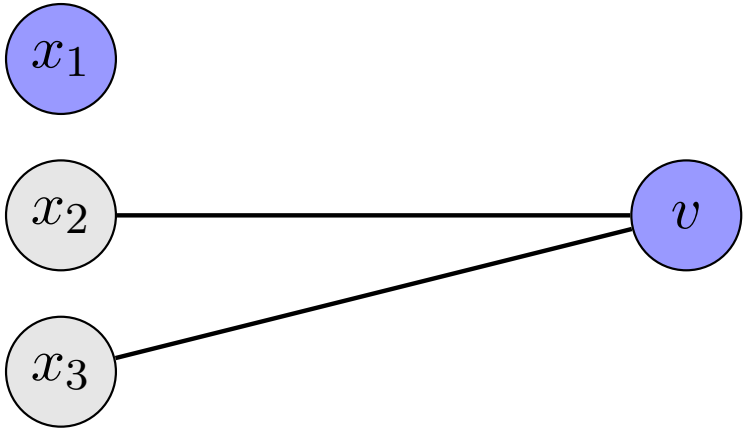
Zwykłe wierzchołki



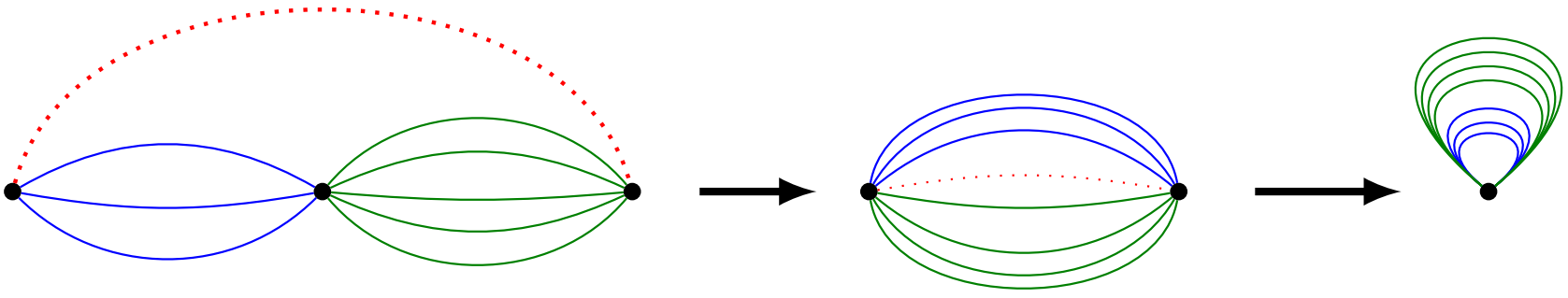


Wierzchołki graniczne





Krawędzie równoległe i pętle



Mikrostruktura

Dla małych grafów z liczbą wierzchołków $t = O(log^4\ log^4\ n)$:

Mikrostruktura

Dla małych grafów z liczbą wierzchołków $t = O(log^4\ log^4\ n)$:

- Tylko operacje łączenia (bez usuwania krawędzi)

Mikrostruktura

Dla małych grafów z liczbą wierzchołków $t = O(log^4\ log^4\ n)$:

- Tylko operacje łączenia (bez usuwania krawędzi)
- Każdy możliwy graf możemy zakodować na $O(t^2)$ bitów

Mikrostruktura

Dla małych grafów z liczbą wierzchołków $t = O(log^4\ log^4\ n)$:

- Tylko operacje łączenia (bez usuwania krawędzi)
- Każdy możliwy graf możemy zakodować na $O(t^2)$ bitów
- Każdy możliwy zbiór wierzchołków granicznych na $O(t)$ bitach

Mikrostruktura

Dla małych grafów z liczbą wierzchołków $t = O(log^4\ log^4\ n)$:

- Tylko operacje łączenia (bez usuwania krawędzi)
- Każdy możliwy graf możemy zakodować na $O(t^2)$ bitów
- Każdy możliwy zbiór wierzchołków granicznych na $O(t)$ bitach
- Każdy możliwy ciąg operacji łączenia na $O(t\ log\ t)$ bitach (max. t operacji)

Mikrostruktura

Dla małych grafów z liczbą wierzchołków $t = O(log^4\ log^4\ n)$:

- Tylko operacje łączenia (bez usuwania krawędzi)
- Każdy możliwy graf możemy zakodować na $O(t^2)$ bitów
- Każdy możliwy zbiór wierzchołków granicznych na $O(t)$ bitach
- Każdy możliwy ciąg operacji łączenia na $O(t\ log\ t)$ bitach (max. t operacji)
- Stąd wszystkich możliwych kodowań jest rzędu $O(2^{poly(t)})$

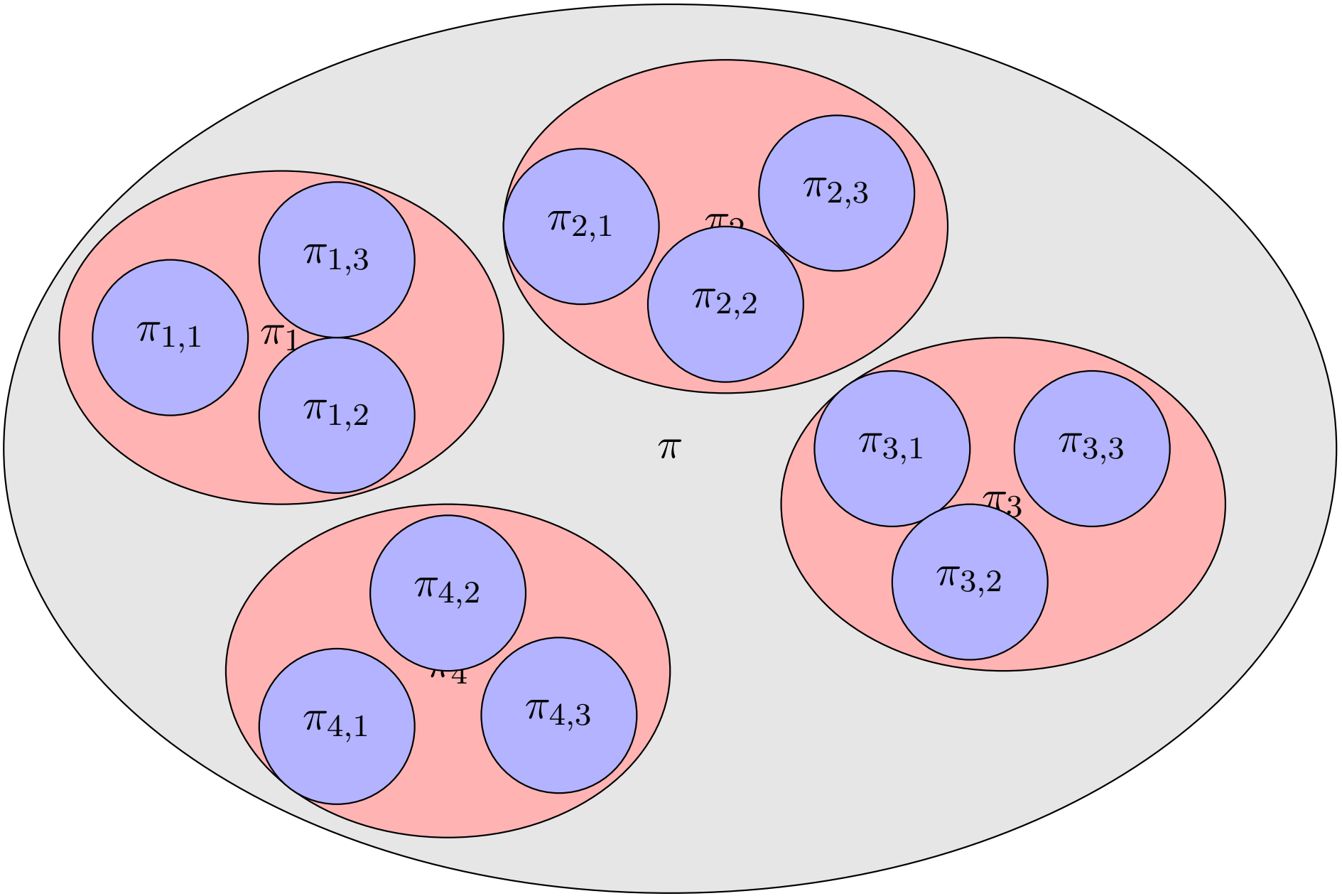
Mikrostruktura

Dla małych grafów z liczbą wierzchołków $t = O(log^4\ log^4\ n)$:

- Tylko operacje łączenia (bez usuwania krawędzi)
- Każdy możliwy graf możemy zakodować na $O(t^2)$ bitów
- Każdy możliwy zbiór wierzchołków granicznych na $O(t)$ bitach
- Każdy możliwy ciąg operacji łączenia na $O(t\ log\ t)$ bitach (max. t operacji)
- Stąd wszystkich możliwych kodowań jest rzędu $O(2^{poly(t)})$

Robimy preprocessing $O(n)$ kodujący na bitach wszystkie możliwe przejścia na wszystkich możliwych grafach.

Wielopoziomowa struktura



Wielopoziomowa struktura

Dla każdego $D \in \Pi$ dodatkowo utrzymujemy:

Wielopoziomowa struktura

Dla każdego $D \in \Pi$ dodatkowo utrzymujemy:

- Dla każdego wierzchołka granicznego, wskaźnik na bliźniaczy wierzchołek w ojcu $par(D)$.

Wielopoziomowa struktura

Dla każdego $D \in \Pi$ dodatkowo utrzymujemy:

- Dla każdego wierzchołka granicznego, wskaźnik na bliźniaczy wierzchołek w ojcu $par(D)$.
- Mapę z pary (D', x), gdzie D' to dziecko D, a x jest interesującym wierzchołkiem dla D, do bliźniaczego wierzchołka granicznego w dziecku D'.