Problem cyklu Eulera w drzewie

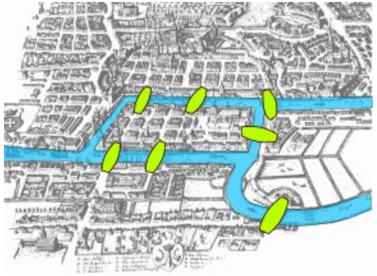
Krzysztof Kamiński
Monika Nogaj
Agata Nielipińska

Historyjka

Zagadnienie mostów królewieckich

W XVIII wieku, w czasach, gdy Euler zajmował się problemem mostów, Kaliningrad, będący wówczas Królewcem, a nazywany również Königsbergiem należał do Królestwa Prus. Wyspy leżące na przepływającej przez miasto Pregole połączone były między sobą i z brzegami rzeki siedmioma mostami w sposób przedstawiony na poniższej grafice.

Problemem, nad którym pracował szwajcarski matematyk było ustalenie trasy wędrówki po Królewcu w taki sposób, by odwiedzić każdy z siedmiu mostów przejść przez niego tylko raz



http://www.wiw.pl/delta/mosty.asp

Wnioski

Na podstawie powyżej zaprezentowanego dowodu-historyjki widać, że cykle Eulera nie występują w drzewach, a jedynie na rzekach. Niestety Leonhard Euler nie podjął próby przejścia drzewa każdą gałąź odwiedzając tylko raz...

Dziękujemy!

Szerokość drzewowa

Algorytmika Problemów Trudnych Obliczeniowo

Krzysztof Kamiński
Monika Nogaj
Agata Nielipińska

Historyjka

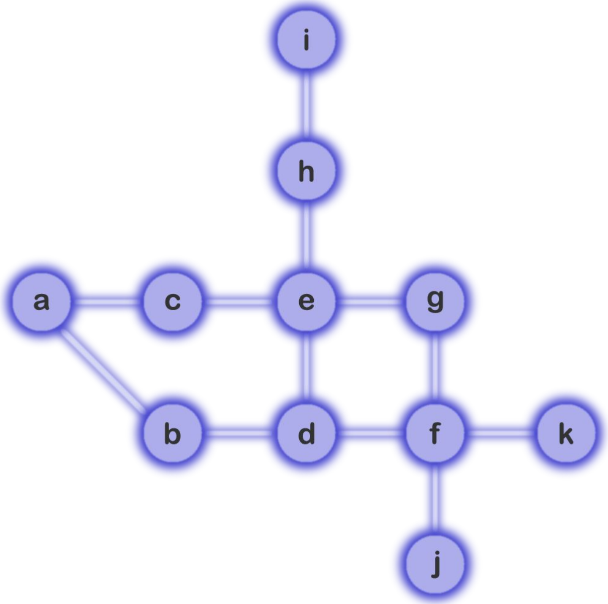
Wyobraźmy sobie, że planujemy przyjęcie i chcemy zaprosić osoby które pracują w naszej firmie. Oczywiście, niektórzy pracownicy są bardziej rozrywkowi od innych i sprawią, że nasza impreza będzie dłużej pamiętana. Jednak nawet najzabawniejsza osoba nie będzie w stanie się dobrze bawić, jeżeli zaprosimy jej bezpośredniego szefa.

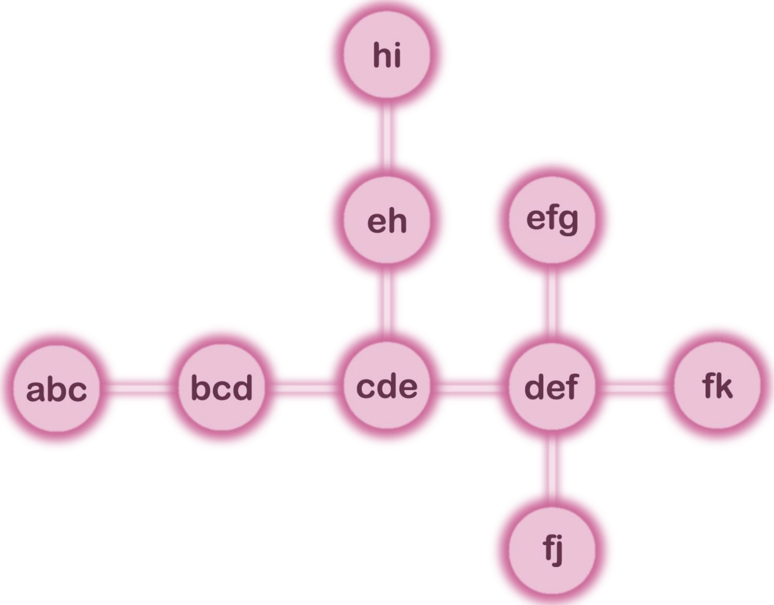
Problem: Zorganizować najlepszą imprezę wszech czasów!

Definicja
Drzewowa dekompozycja grafu

Dla grafu G dekompozycją drzewową nazwiemy takie drzewo T i rodzinę zbiorów wierzchołków (Vt), że:

- $V(G) = \bigcup_t Vt$
- dla każdej krawędzi $uv \in E(G)$ istnieje t, że $u,v \in Vt$
- dla każdego $v \in V(G)$ zbiór $\{t : v \in Vt\}$ jest spójny w T


i
h
a
c
e
g
b
d
f
k
j


hi
eh
efg
abc
bcd
cde
def
fk
fj

Szerokość dekompozycji (width): $\max_{t \in T} |V_t| - 1$

Szerokość drzewowa (treewidth): min(possible width)

Ścieżkowa dekompozycja grafu

To jest to samo co drzewowa, tylko robimy ścieżkę a nie drzewo.

Nice path decomposition

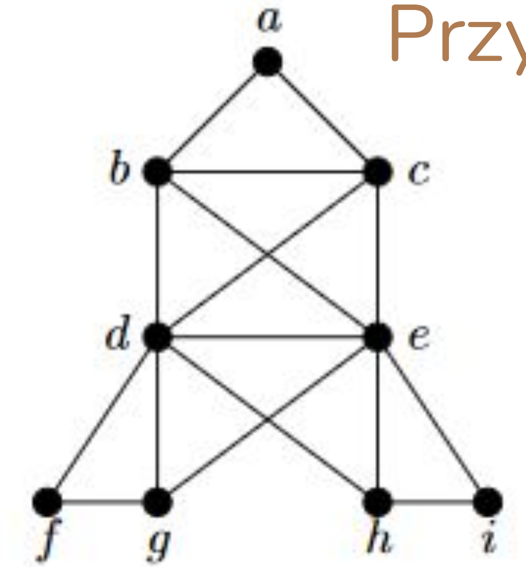
To ścieżkowa dekompozycja P= $(X1,X2,\ldots,Xr)$ grafu, taka że ˆ$|X_1|=|X_r|=1$ oraz dla każdego $i\in \{1,2,\ldots,r-1\}$

E wierzchołek $v \notin X_i$ oraz $X_{i+1}=X_i \cup \{v\}$

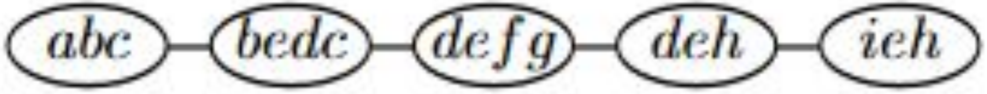
lub

E wierzchołek $w \in X_i$ oraz $X_{i+1}=X_i \setminus \{w\}$

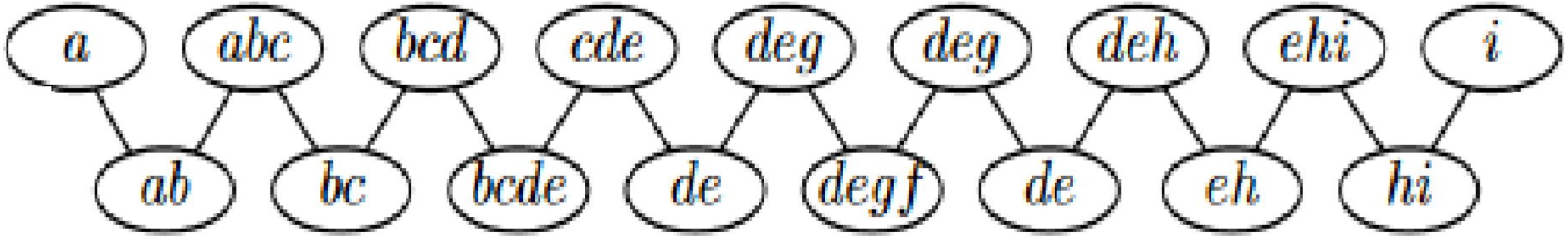
Przykład- graf Rakieta



path decomposition



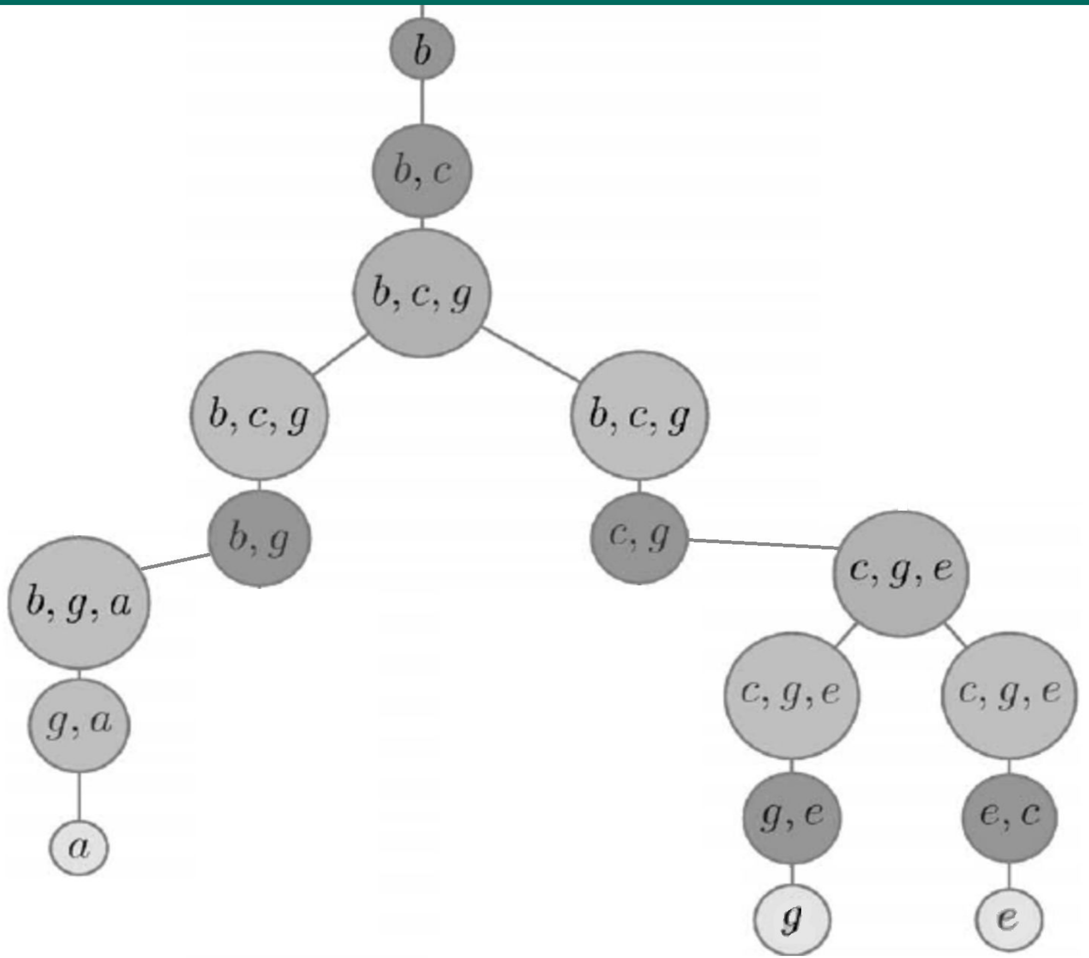
nice path decomposition



Nice tree decomposition

To drzewowa dekompozycja T grafu G, w której wyróżniony został korzeń, każdy liść zawiera dokładnie jeden wierzchołek z G oraz każdy pozostały wierzchołek $t \in V(T)$ jest jednym z 3 rodzajów:

- Introduce node: t ma jedno dziecko: t'
  E $v \in V(G)$ taki że $t = t' \cup \{v\}$
- Forget node: t ma jedno dziecko: t'
  E $v \in V(G)$ taki że $t = t' \setminus \{v\}$
- Join node: t ma dwoje dzieci t' i t'' oraz $t = t' = t''$


b
b, c
b, c, g
b, c, g
b, c, g
b, g
c, g
c, g, e
b, g, a
c, g, e
c, g, e
g, a
g, e
e, c
a
g
e

Algorytmy znajdowania dekompozycji drzewowej grafu

Problem określenia szerokości drzewowej jest NP-trudny, jednak z racji szerokiego zastosowania poszukiwane i rozwijane są algorytmy pozwalające określić szerokość drzewową oraz zwracające odpowiednią dekompozycję drzewową.

Algorytmy znajdowania dekompozycji drzewowej grafu

Jednym ze sposobów jest podejście wykorzystujące metodę podziału i ograniczeń. Metoda polega na eliminacji kolejnych wierzchołków z grafu G i dodawaniu do drzewa T węzłów zawierających te wierzchołki wraz z ich wszystkimi sąsiadami.

Problem: n! różnych możliwości uszeregowania wierzchołków

Algorytmy znajdowania dekompozycji drzewowej grafu

Gogate i Dechter zaproponowali w swojej pracy kilka zasad według których wyznaczyć granice i dorzucić ścieżki, które z największym prawdopodobieństwem nie prowadzą do wyznaczenia optymalnego rozwiązania.

Ich algorytm gwarantuje dość dobre górne granice dokładności zwróconego rozwiązania.

Algorytmy znajdowania dekompozycji drzewowej grafu

Algorytmy heurystyczne skupiają się głównie na wyznaczaniu najlepszego porządku eliminacji wierzchołków.

Czasem też korzysta się z metody podziału i ograniczeń, przerywając działanie algorytmu w określonym momencie i zwracając najlepszy do tej pory znaleziony wynik.

Algorytmy znajdowania dekompozycji drzewowej grafu

Częściej rozważanym problemem jest znalezienie dekompozycji drzewowej dla grafu G o ograniczonej szerokości drzewowej k.

Algorytmy takie dla zadanego grafu G i stałej w:

- jeśli szerokość drzewowa G < w zwracają dekompozycję grafu G
- w przeciwnym przypadku informację, że szerokość drzewowa G jest co najmniej równa w

Algorytmy znajdowania dekompozycji drzewowej grafu

Zaprezentujemy algorytm, który dla G oraz w, takich, że szerokość drzewowa G mniejsza od w, zwraca dekompozycję drzewową grafu G o szerokości mniejszej niż 4w.

złożoność: O(f(w)mn)

gdzie m - liczba krawędzi, n - liczba wierzchołków

Algorytmy znajdowania dekompozycji drzewowej grafu

Twierdzenie 1

Niech $G=(V,E)$ ma m krawędzi, X będzie podzbiorem jego wierzchołków o liczności k oraz $w \le k$ będzie zadanym parametrem. Możemy sprawdzić czy X jest w-spójny w czasie $O(f(k)m)$. Dodatkowo, jeżeli X nie jest w-spójny, możemy otrzymać dowód w postaci zbiorów ($Y,Z \subset X$) oraz ($S \subset V$) takich że $|S|<|Y|=|Z|\le w$ i nie istnieje ścieżka z $Y-S$ do $Z-S$ w $G-S$.

Algorytmy znajdowania dekompozycji drzewowej grafu

Twierdzenie 2

Jeżeli graf G zawiera (w+1)-spójny podzbiór wierzchołków X o rozmiarze co najmniej 3w, wówczas G ma szerokość drzewową przynajmniej w.

Algorytmy znajdowania dekompozycji drzewowej grafu

Definicja

Jeżeli C jest spójnym podgrafem grafu G-U to każdy wierzchołek u ∈ U jest sąsiadem C jeżeli istnieje wierzchołek v ∈ C oraz krawędź (v,u) ∈ E.

Algorytmy znajdowania dekompozycji drzewowej grafu

Niezmienniki algorytmu:

1. W dowolnym momencie wykonania algorytmu, każdy spójny podgraf C grafu G-U ma co najwyżej 3w sąsiadów oraz istnieje V t zawierająca wszystkich sąsiadów C.
2. W dowolnym momencie działania algorytmu skonstruowane drzewo jest częściowym drzewem dekompozycji o rozmiarze mniejszym niż 4w (czyli jeżeli drzewo pokrywa U wierzchołków to jest to drzewo dekompozycji dla podgrafu G złożonego z wierzchołków z U).

1. Rozpoczynamy z pustym drzewem dekompozycji ($U = \varnothing$).
2. Dopóki G-U jest niepusty dodajemy nowy wierzchołek do drzewa dekompozycji:

   T - drzewo dekompozycji

   C - spójna składowa grafu G-U

   X - zbiór sąsiadów C

   t - wierzchołek zawierający cały zbiór X

   Możliwe sytuacje:

   a) zbiór X ma mniej niż 3w elementów
   b) zbiór X ma dokładnie 3w elementów

a) zbiór X ma mniej niż 3w elementów

Bierzemy dowolny $v \in C$ dla którego istnieje krawędź do pewneo wierzchołka z X i do drzewa T dodajemy węzeł $s = (X \cup \{v\})$ oraz krawędź (s, t)

- $|s| = |X| + 1 <= 3w$
- wszyscy sąsiedzi v z U są w X, więc
  $T` = (V(T) \cup \{s\}, E(T) \cup \{(s,t)\})$ jest drzewem dekompozycji grafu G indukowanego przez zbiór $U \cup \{v\}$

Zatem niezmienniki algorytmu pozostają zachowane

b) zbiór X ma dokładnie 3w elementów

Sprawdzamy czy X jest (w+1)-spójny (korzystając z tw. 1)

- jeśli jest, to na mocy tw. 2 G ma szerokość drzewową równą co najmniej w. To kończy algorytm.
- w przeciwnym przypadku z tw. 1 otrzymujemy ($Y,Z \subset X$) oraz ($S \subset V$) takich że $|S|<|Y|=|Z|<= w$ i nie istnieje ścieżka z Y-S do Z-S w G-S.

  Wówczas niech $S` = S \cap (Y \cup Z \cup C)$ oraz $S` \cap C$ nie może być zbiorem pustym. Ponadto $|S`|<=|S|<w$.

  Do drzewa dodajemy nowy węzeł $s = (X \cup S`)$ oraz krawędź (s,t)

- C może rozpaść się na kilka spójnych składowych. Rozpatrując dowolną składową C\` zauważamy, że każdy jej sąsiad należy do s oraz wszyscy jej sąsiedzi należą albo do ((X-Z) ∪ S\`) albo do ((X-Y) ∪ S\`) i każdy z tych zbiorów zawiera maksymalnie 3w wierzchołków.
- |S\`| < w i |X| = 3w, a więc |s| = |S\` ∪ X| <= 4w, oraz T\` = (V(T) ∪ {s}, E(T) ∪ {(s,t)}) jest drzewem dekompozycji grafu G indukowanego przez zbiór U ∪ {S\`} o szerokości drzewowej nie większej niż 4w

Zatem niezmienniki algorytmu pozostają zachowane

Przedstawiony algorytm w każdym kroku zmniejsza liczbę wierzchołków grafu minimum o 1 a więc skończy się po maksymalnie n krokach.

Złożoność algorytmu:

- sprawdzanie (w+1) spójności: O(f(w)m)
- liczba kroków algorytmu: O(n)
- sumaryczna złożoność: O(f(w)mn)

Funkcja f(w) jest funkcją wykładniczą rzędu 2^(6w)

Algorytm znajdowania największego zbioru niezależnego

Dekompozycja drzewowa bez powtórzeń

Dekompozycja drzewowa T grafu G jest dekompozycją bez powtórzeń wtedy i tylko wtedy gdy dla żadnej krawędzi (x,y) z drzewa nie zachodzi ($V_x \subset V_y$).

Każdą dekompozycję drzewową można przekształcić w dekompozycję bez powtórzeń.

Własności drzewowej dekompozycji grafów

Własność I

Niech las powstały w wyniku usunięcia wierzchołka t z drzewa T (T-t) składa się z drzew: T_1 ,...,T_d , wówczas podgrafy: G_{T1} -V_t , G_{T2} -V_t ,...,G_{Td} -V_t nie mają wspólnych wierzchołków i nie ma pomiędzy nimi żadnych krawędzi.

V_t - zbiór wierzchołków, które tworzą wierzchołek t w dekompozycji drzewnej.

Własność II

Niech X i Y będą dwoma drzewami powstałymi w wyniku usunięcia krawędzi (x,y) z T. Wówczas usunięcie ($V_x \cap V_y$) ze zbioru wierzchołków grafu rozspójnia G na dwa podgrafy:

G_x - ($V_x \cap V_y$) oraz G_y - ($V_x \cap V_y$).

Własności drzewowej dekompozycji grafów

Własność III

Każda dekompozycja drzewowa bez powtórzeń grafu o n wierzchołkach zawiera nie więcej niż n elementów.

Problem zbioru niezależnego - intuicja

Optymalny zbiór niezależny zawiera część wierzchołków z V_t, ale nie wiemy jakie to wierzchołki. Oznaczmy optymalny zbiór przez U. Sprawdzimy wszystkie możliwości, czyli 2^{w+1}, ponieważ V_t może mieć maksymalnie w+1 wierzchołków. (w - szerokość drzewowa). Dzięki własnościom I i II w dwóch różnych poddrzewach problemy można rozwiązywać niezależnie.

Algorytm

Definicja

Niech podzbiór U zbioru V_t reprezentuje przecięcie optymalnego rozwiązania z V_t. Dla każdego niezależnego zbioru ($U \subset V_t$) definiujemy:

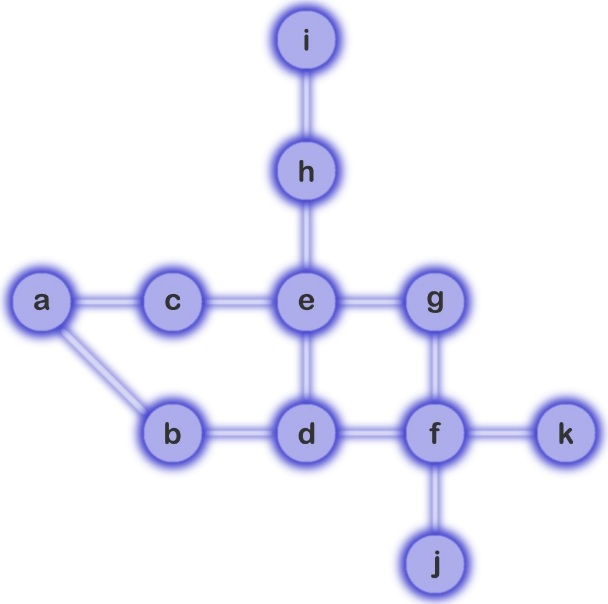
$$F_t(U) = max\ |S|,\ gdzie\ S\ to\ zbiór\ niezależny\ w\ G_t\ taki\ że\ (S \cap V_t) = U$$

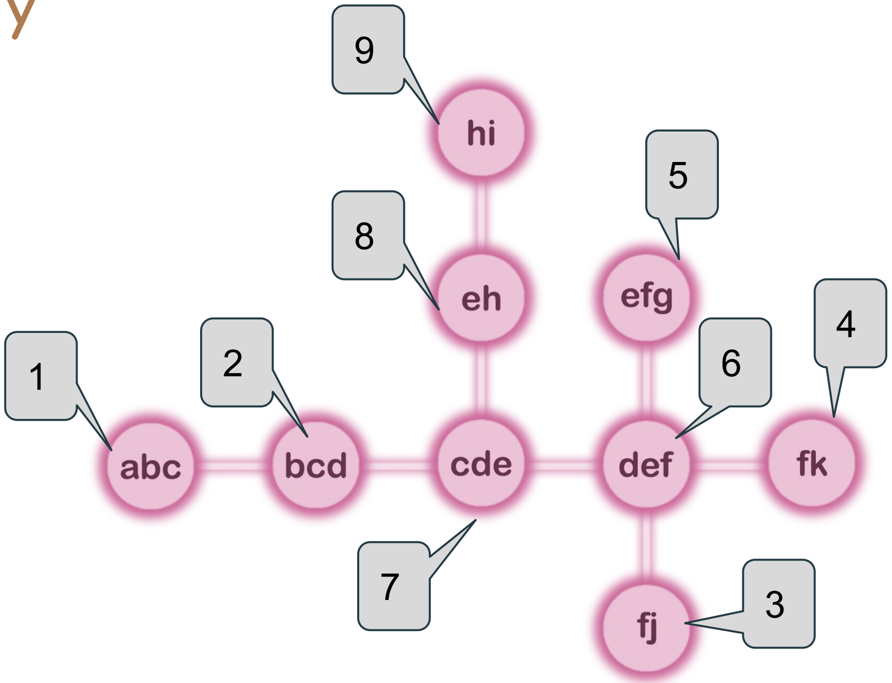
$F_t(U)$ jest niezdefiniowany jeżeli U nie jest zbiorem niezależnym.

```
T = drzewo dekompozycji grafu G bez powtórzeń
for (t – wierzchołki T w kolejności postorder) {
        if (t jest liściem) {
                for (U – niezależne zbiory w V_t) {
                        F_t(U) = U
                }
        } else {
                for (U – niezależne zbiory w V_t) {
                        F_t(U) = U ∪ ⋃_{i ∈ potomkowie t} Z_i
                }
        }
}
```

$Z_i = \max\{F_{ti}(U_i) \setminus (U_i \cap U) : (U_i \cap V_t = U \cap V_{ti})$ i $U_i \subseteq V_{ti}$ jest niezależny $\}$

Przykład na tablicy





Analiza złożoności

- sprawdzenie czy $U_i \cap V_t = U \cap V_{ti}$: **O(w)**
- liczba zbiorów niezależnych w V_{ti}: $\mathbf{O(2^{w+1})}$
- liczba generowanych zbiorów niezależnych w każdym węźle: $\mathbf{O(2^{w+1})}$
- liczba węzłów: **O(n)**

- sumaryczna złożoność: $\mathbf{O(2^{2(w+1)}wn)}$

Dziękujemy !