Algorytmy aproksymacyjne

Metoda Baker

inż. Filip Grajek, inż. Jakub Mazurkiewicz, inż. Michał Omelańczuk

Plan prezentacji

1. Wprowadzenie i definicje
2. Ogólny schemat algorytmu
3. Największy zbiór niezależny dla grafu zewnętrznie planarnego
4. Największy zbiór niezależny dla grafu k-zewnętrznie planarnego
5. Złożoność i jakość algorytmu

Algorytmy aproksymacyjne

- Algorytm znajdujący przybliżone rozwiązanie problemów optymalizacyjnych, stosuje się je dla problemów dla których nie są znane szybkie algorytmy znajdujące rozwiązanie optymalne.

- Jeżeli istnieje stała a≥1 taka że dla każdego zadania rozwiązywanego problemu algorytm znajduje rozwiązanie nie gorsze niż
a * rozwiązanie optymalne to taki algorytm nazywany a-aproksymacyjnym.
- Jeżeli istnieje funkcja α(n) taka że α(n) ≥1 dla każdego n i dla każdego zadania tego problemu znajduje rozwiązanie nie gorsze niż α(n) * optymalne rozwiązanie to taki algorytm nazywamy α-aproksymacyjnym

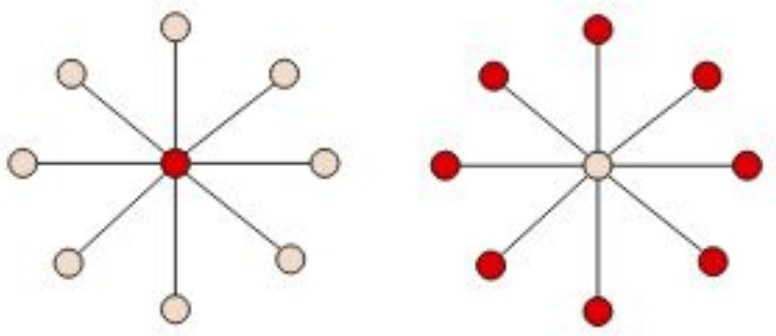
Schemat aproksymacyjny

- Po dodaniu do algorytmu aproksymacyjnego dodatkowego parametru **ε** otrzymujemy schemat aproksymacyjny.

- Schemat aproksymacyjny przyjmuje na wejściu problem optymalizacyjny oraz **ε** i znajduje rozwiązanie które jest nie gorsze niż (1+ε) optymalnego.

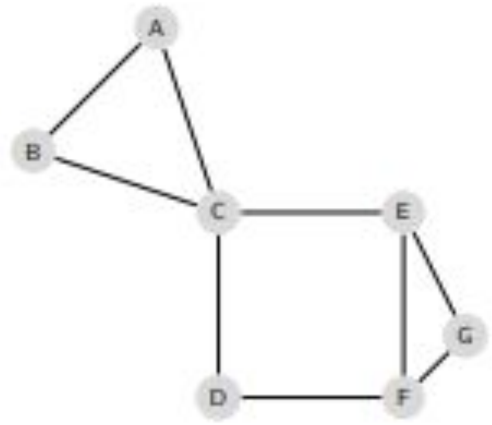
Problem maksymalnego zbioru niezależnego w grafie - maximum independent set problem

- Największy zbiór niezależny to największy zbiór wierzchołków w grafie taki że nie ma krawędzi pomiędzy żadną parą wierzchołków.
- Jest to problem NP-trudny



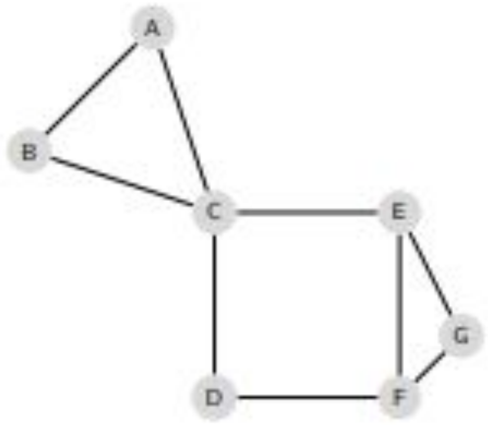
Graf planarny

Graf planarny - który można narysować na płaszczyźnie tak aby krawędzie tego grafu nie przecinały się.



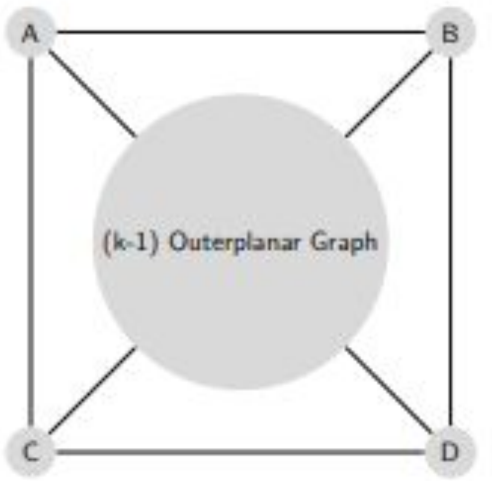
1-outerplanar

1-outerplanar - graf zewnętrznie planarny - to taki graf w którym krawędzie się nie przecinają i wszystkie wierzchołki są na ścianie zewnętrznej (outer face)

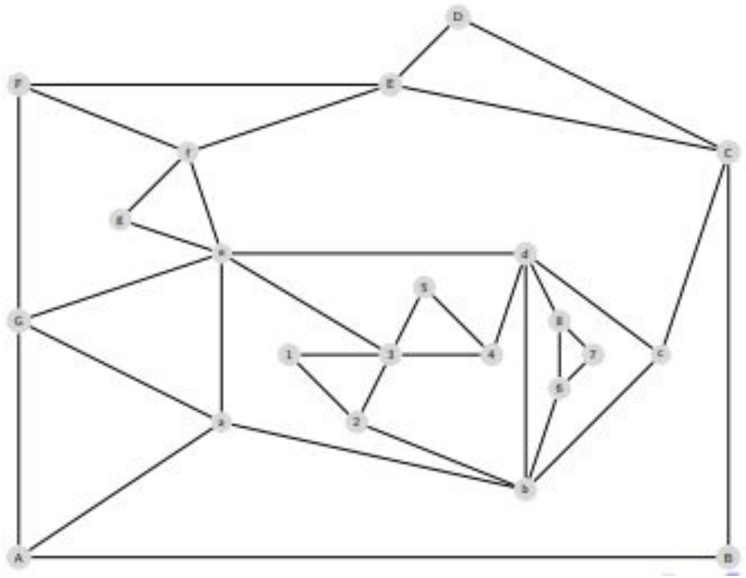


k-outerplanar

k-outerplanar - taki graf w którym krawędzie się nie przecinają i po usunięciu wierzchołków ze ściany zewnętrznej otrzymujemy graf (k-1)-outerplanar

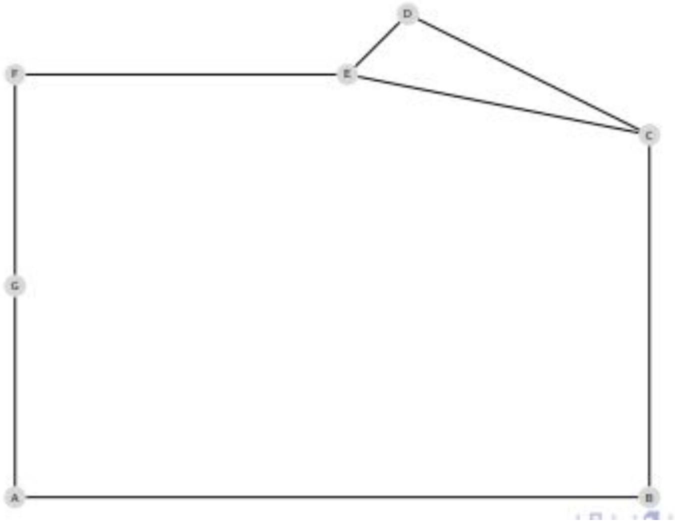


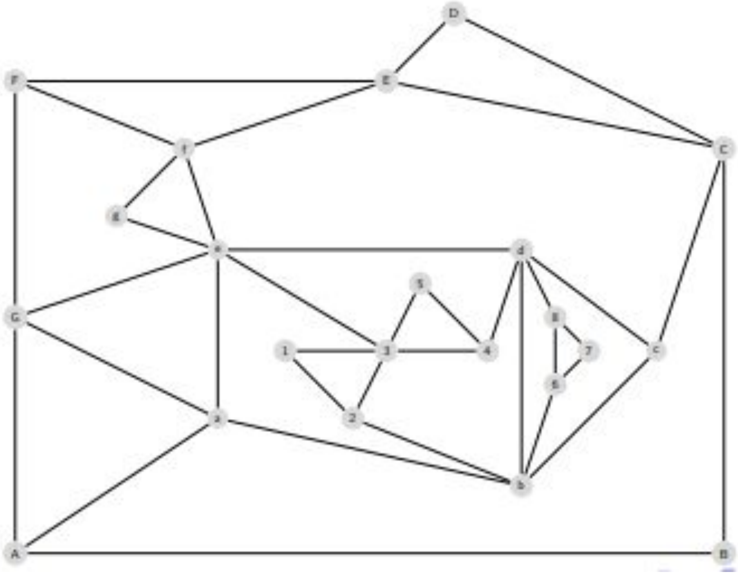
Wierzchołki na warstwie k



Wierzchołki na warstwie 1

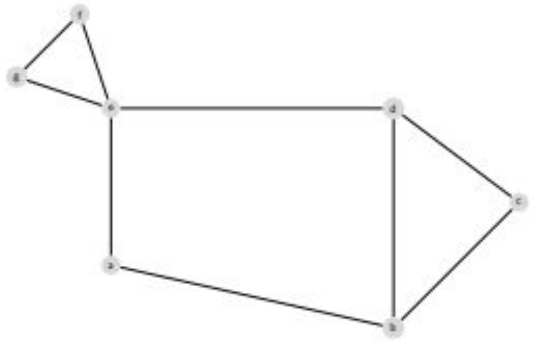
Wierzchołki na warstwie 1 są to wierzchołki ze ściany zewnętrznej

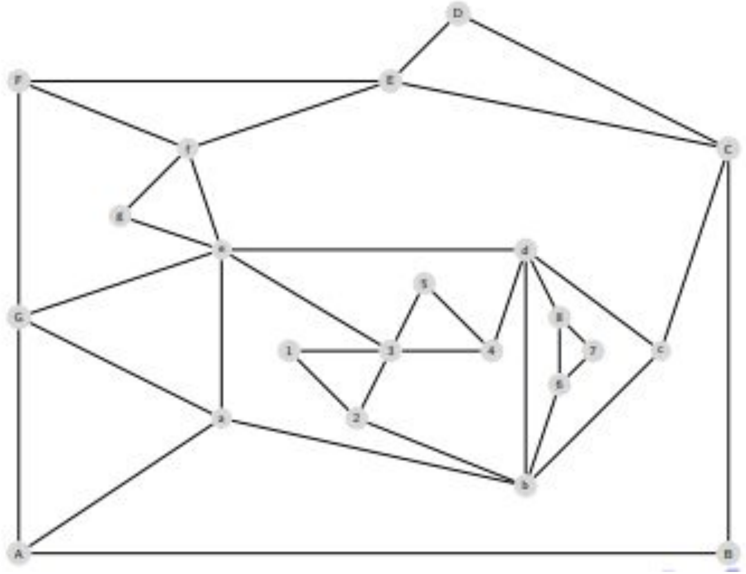




Wierzchołki na warstwie 2

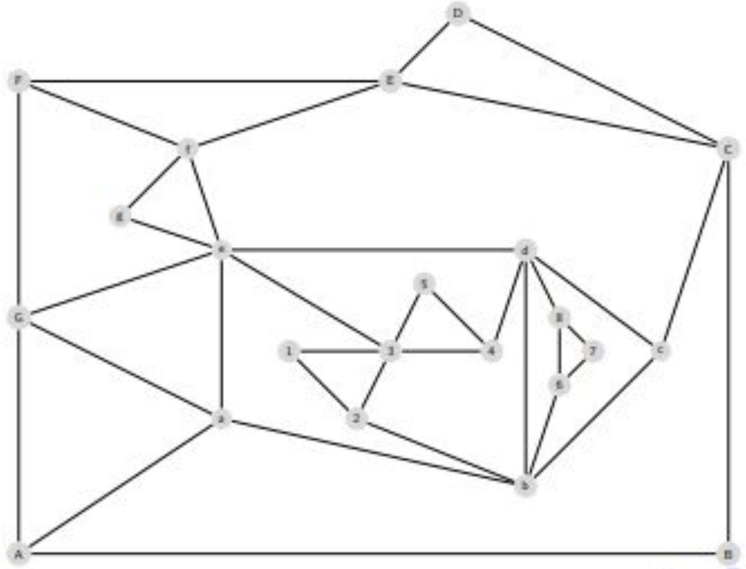
Wierzchołki na warstwie 2 - wierzchołki ze ściany zewnętrznej po usunięciu wierzchołków z 1 warstwy





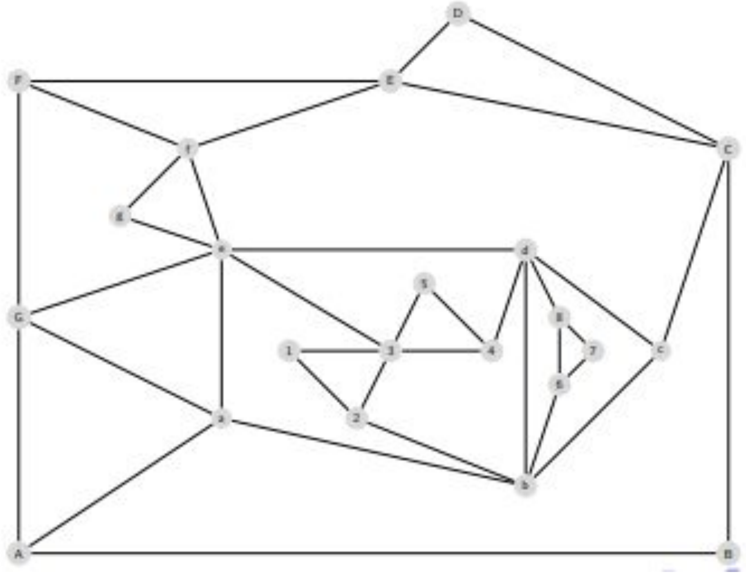
Wierzchołki na warstwie k

Wierzchołki na warstwie k - wierzchołki ze ściany zewnętrznej po usunięciu wierzchołków z warstw od 1 do k-1



Obliczenie warstwy każdego wierzchołka

Obliczenie warstwy każdego wierzchołka jest możliwe w czasie liniowym względem liczby wierzchołków.



Reprezentacja grafu planarnego

Graf planarny reprezentacja - dla każdej krawędzi z grafu tworzymy tablicę z jej wierzchołkami i wskaźnikami na 4 krawędzie sąsiadujące, po 2 dwie dla obu wierzchołków, jedna zgodnie z kierunkiem wskazówek zegara i jedna przeciwnie.

E1	1	3	E2	E2	E5	E4
E2	1	2	E1	E1	E3	E3
E3	2	3	E2	E2	E5	E4

Metoda Brendy Baker

Metoda Brendy Baker służy do aproksymowanego rozwiązywania problemu największego zbioru niezależnego w grafie planarnym.
Znalezione rozwiązanie jest nie gorsze niż k/(k+1) optymalnego rozwiązania o złożoności czasowej liniowej względem liczby wierzchołków w grafie.

Chcemy rozwiązać problem maksymalnego zbioru niezależnego dla podgrafów które są k-outerplanar i ignorować część wierzchołków, a następnie połączyć podgrafy zachowując warunek na zbiór niezależny.

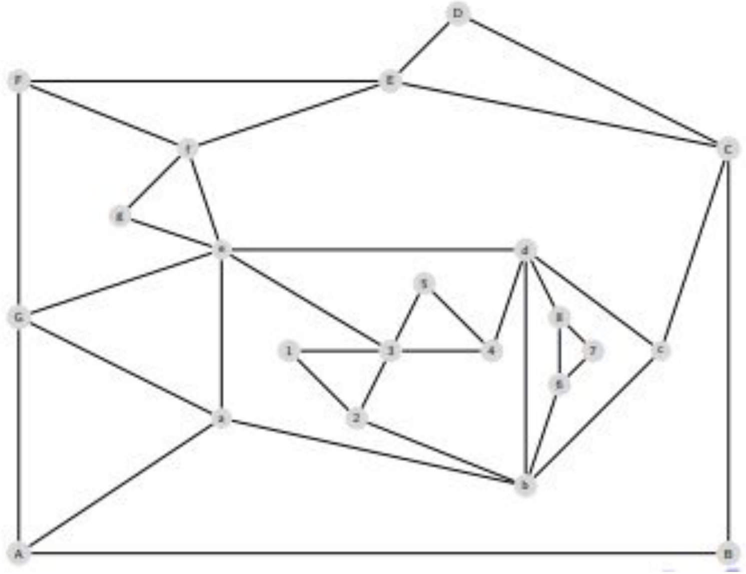
Metoda Brendy Baker

Lista kroków:

1. Liczymy warstwę każdego wierzchołka w grafie.
2. Dla każdego **i** takiego że 0 ≤ **i** ≤ **k** wykonaj:
   a. Usuń wierzchołki z warstw takich że **numer_warstwy mod (k+1)** = **i**, tym samym dzieląc graf na podgrafy k-zewnętrznie planarne
   b. Znajdź maksymalny zbiór niezależny w każdym podgrafie i weź sumę ich rozwiązań. Wzięcie sumy jest możliwe ze względu na to że każde dwa podgrafy są rozłączne.
3. Weź najlepsze rozwiązanie jako rozwiązanie dla całego grafu

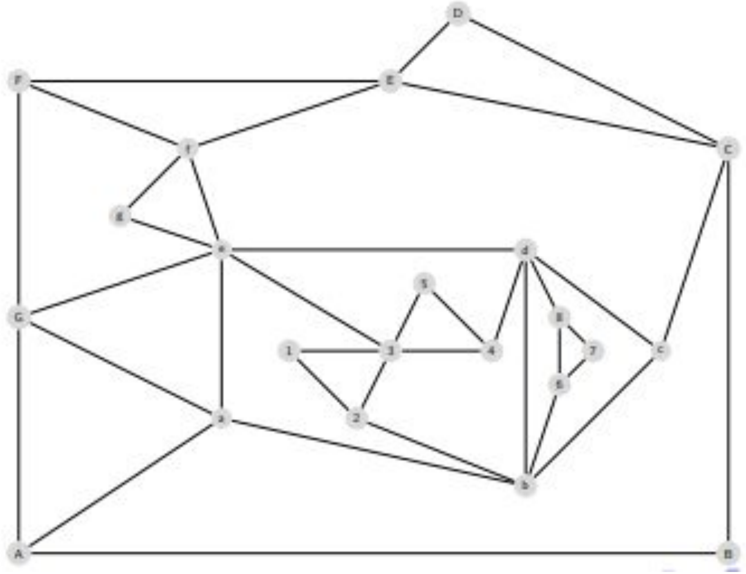
Metoda Brendy Baker

Wejście metody: reprezentacja grafu planarnego oraz parametr k.



Metoda Brendy Baker

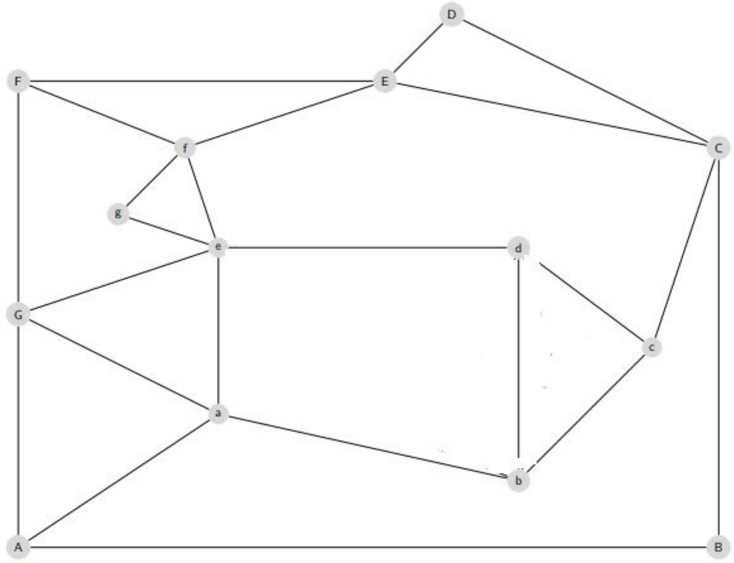
Przykłady dla k=2 oraz dla k =1

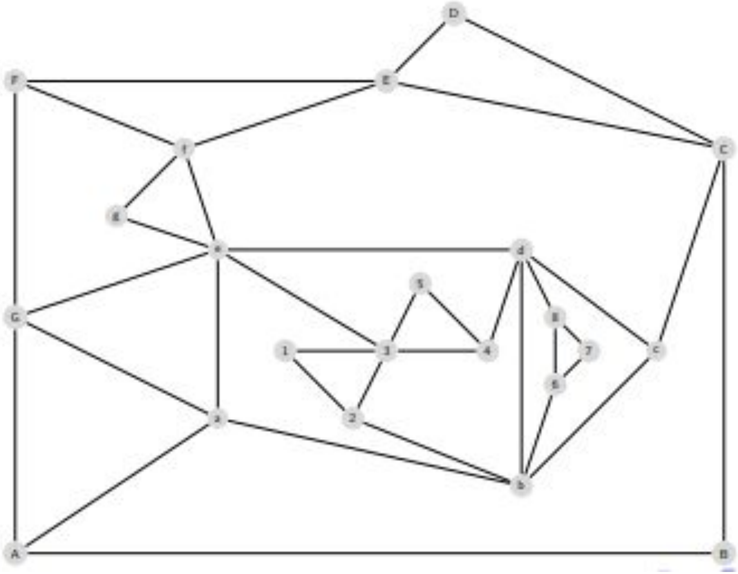


Metoda Brendy Baker

Przykład dla k=2 Iteracja i = 0

numer_warstwy mod (2+1) = **0**

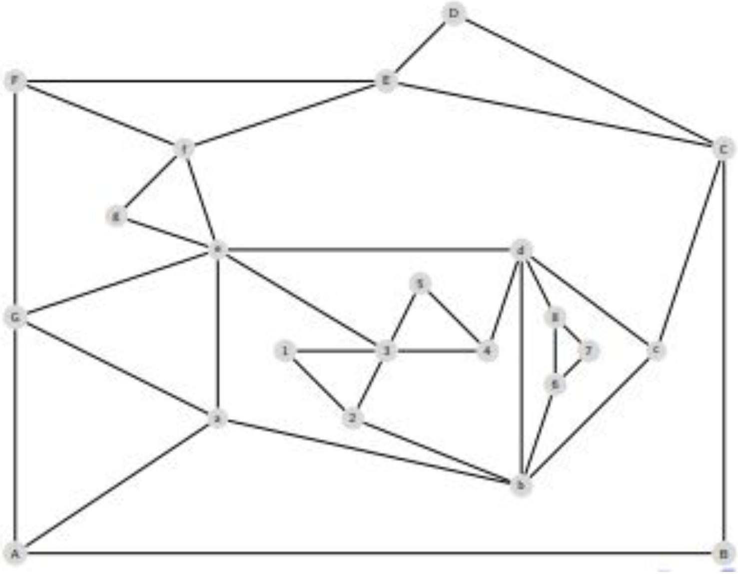




Metoda Brendy Baker

Przykład dla k=2 Iteracja i = 1

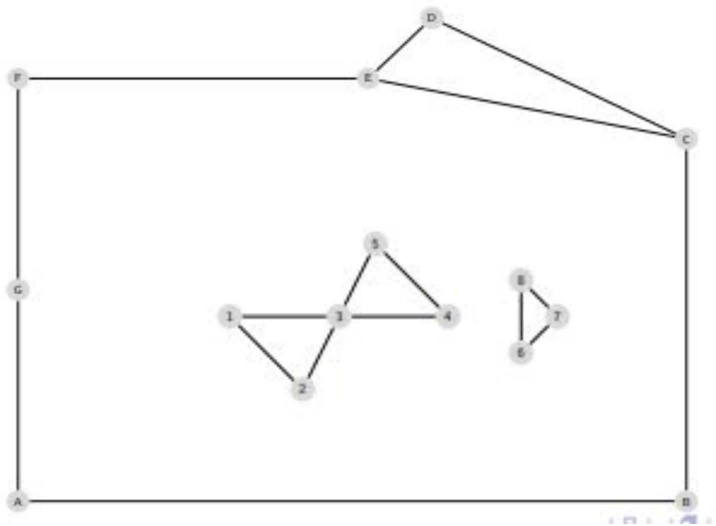
numer_warstwy mod (2+1) = **1**,

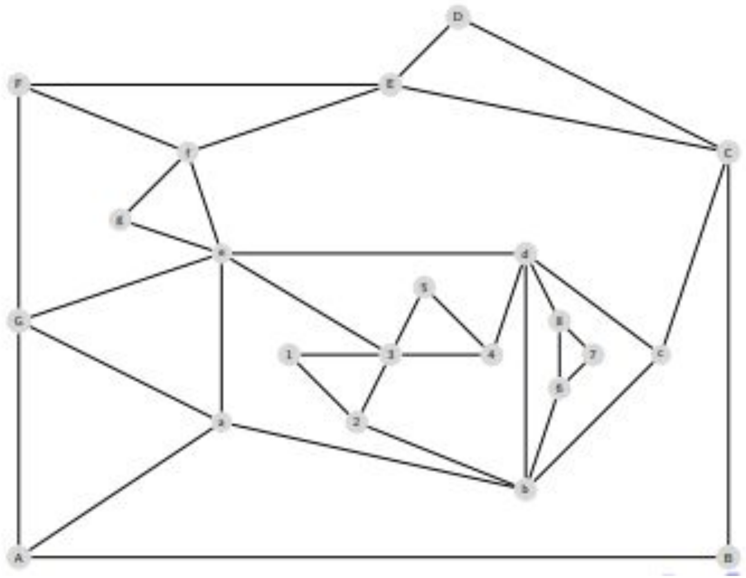


Metoda Brendy Baker

Przykład dla k=2 Iteracja i = 2

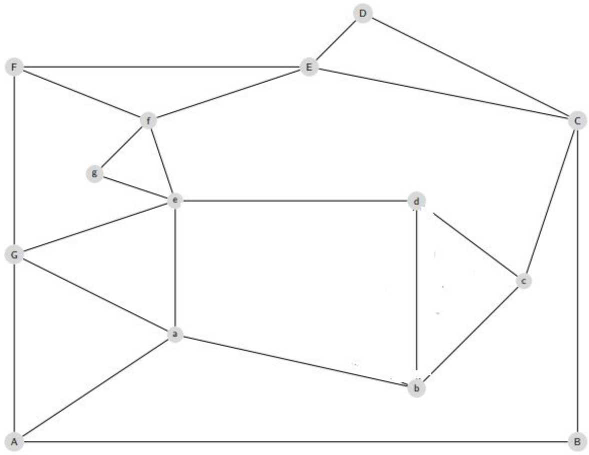
numer_warstwy mod (2+1) = **2**,

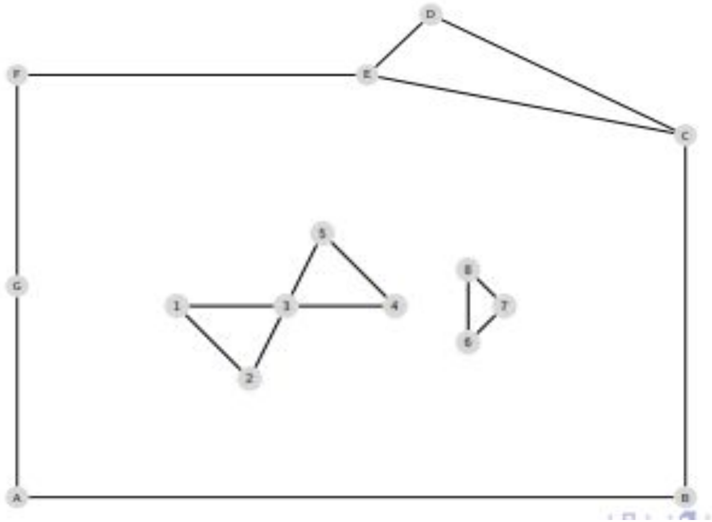




Metoda Brendy Baker

Rozwiązanie

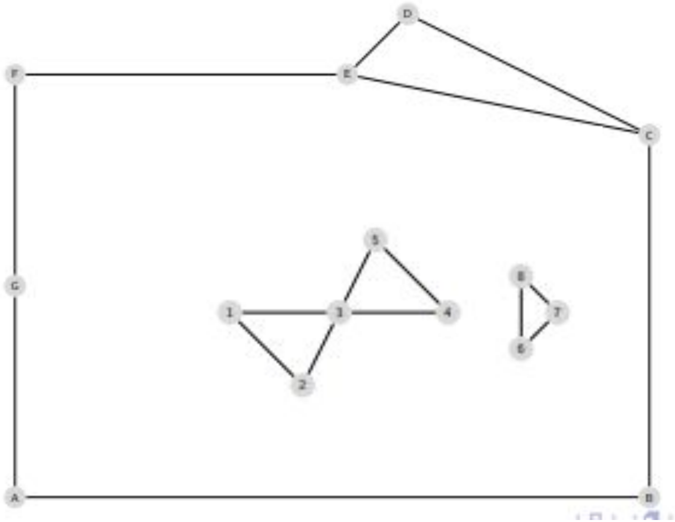


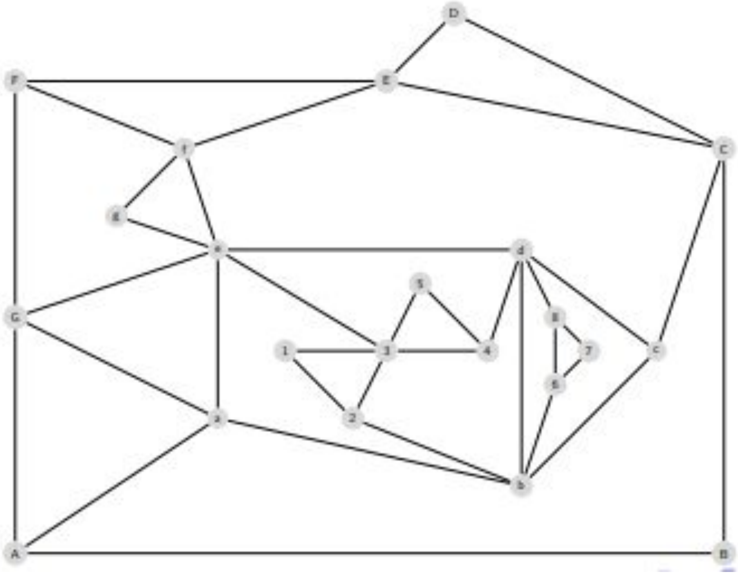


Metoda Brendy Baker

Przykład dla k=1 Iteracja i = 0

numer_warstwy mod (1+1) = **0**,

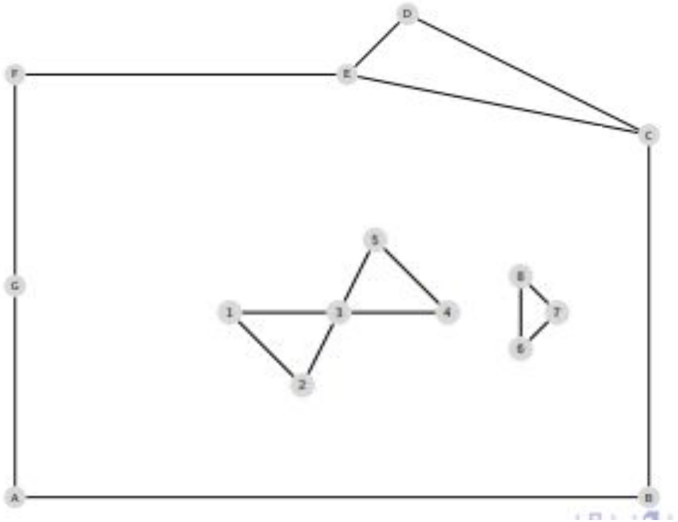


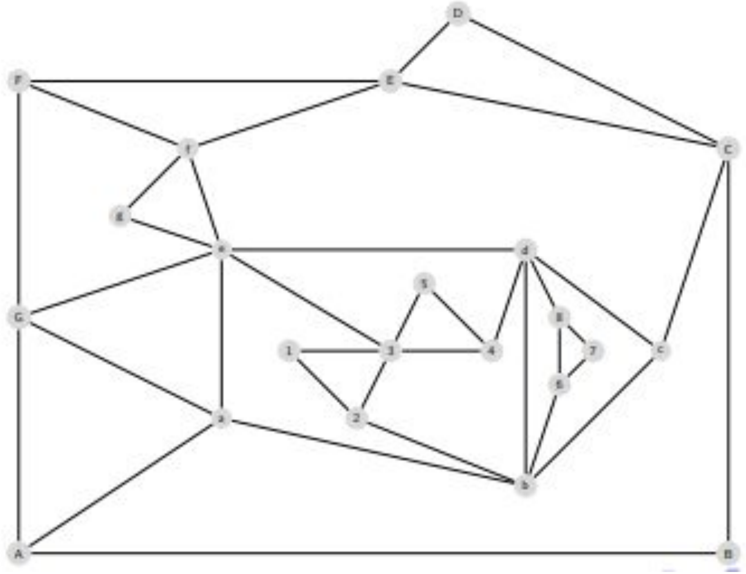


Metoda Brendy Baker

Przykład dla k=1

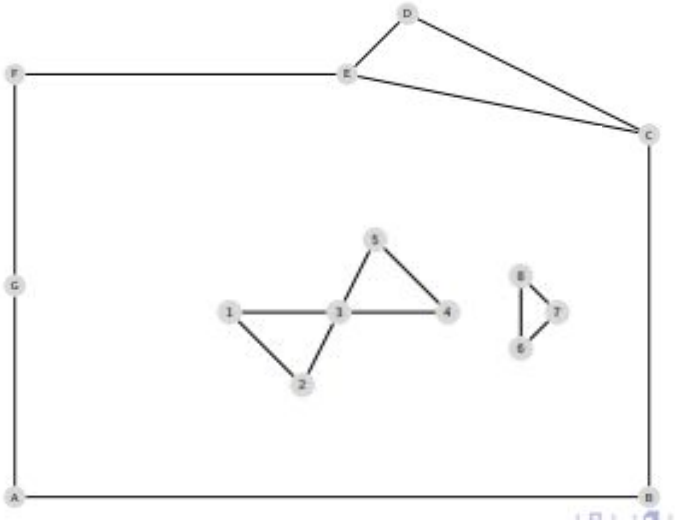
Iteracja i = 1

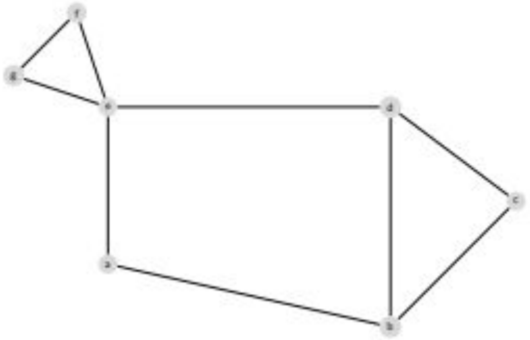




Metoda Brendy Baker

Rozwiązanie





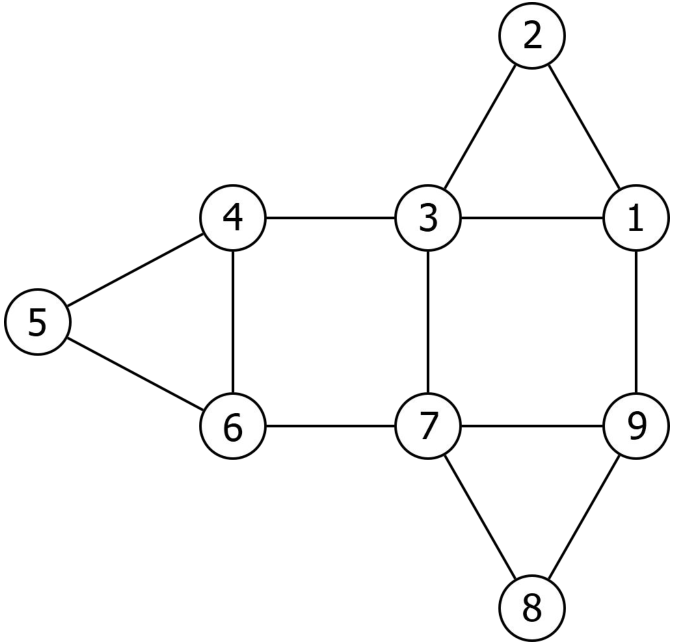
Algorytm dla grafu zewnętrznie planarnego

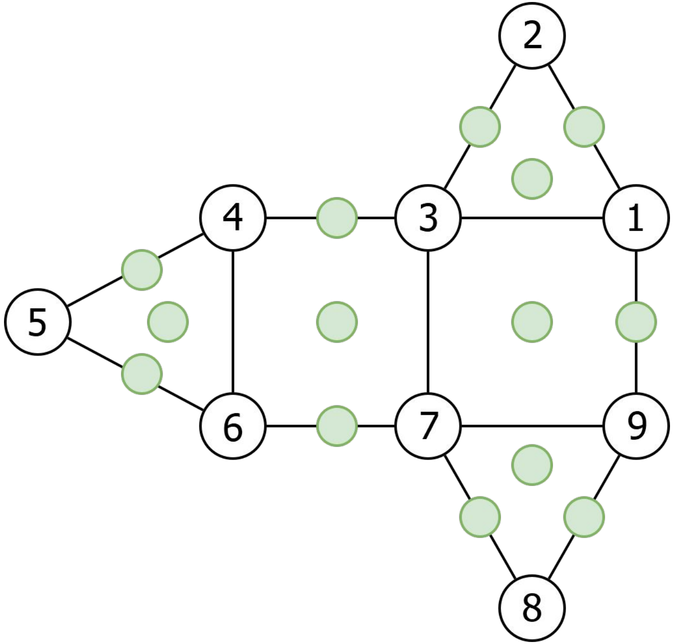
1. Przetwarzanie wstępne
2. Budowa drzewa
3. Analiza drzewa

Przetwarzanie wstępne

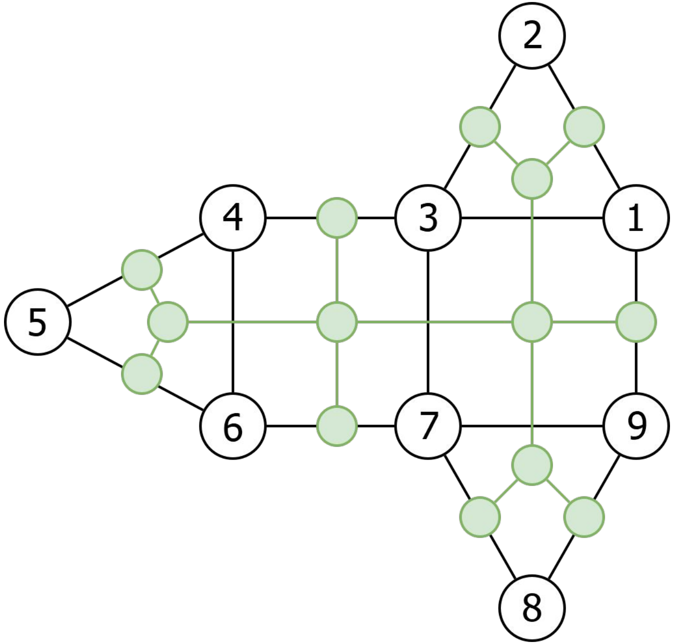
- Graf jest zewnętrznie planarny i spójny
- Każdy most (x, y) należy zamienić na dwie krawędzie (x, y)

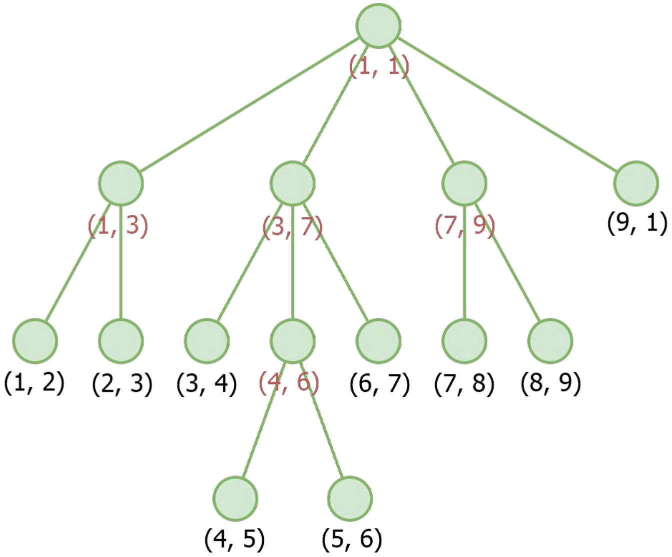
Budowa drzewa dla grafu bez punktów artykulacji



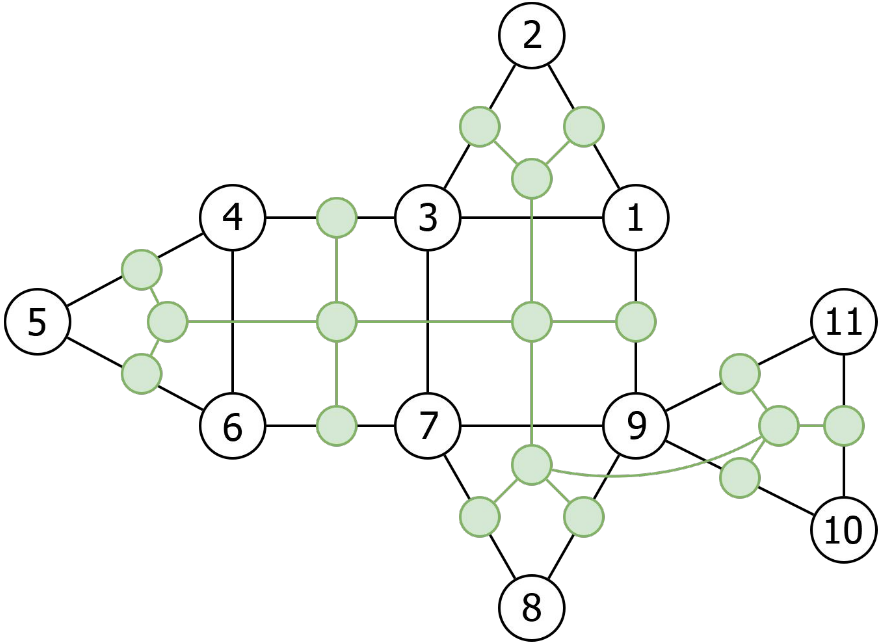


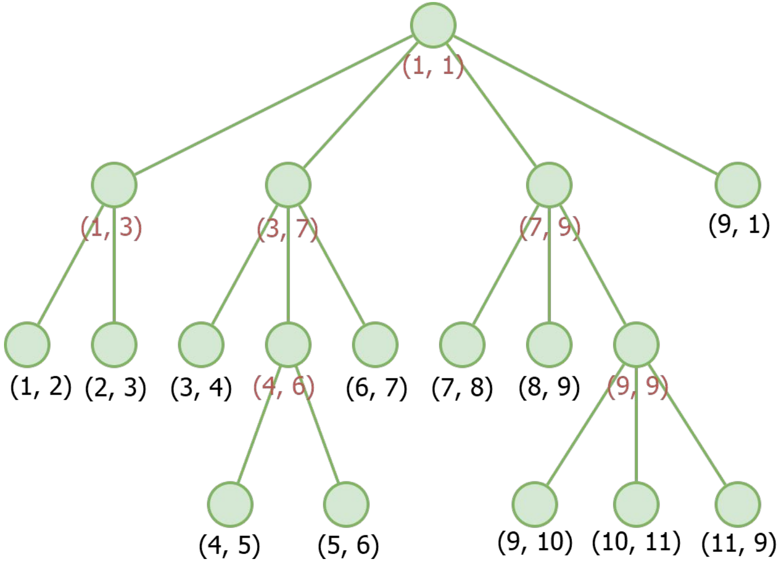
Budowa drzewa dla grafu bez punktów artykulacji





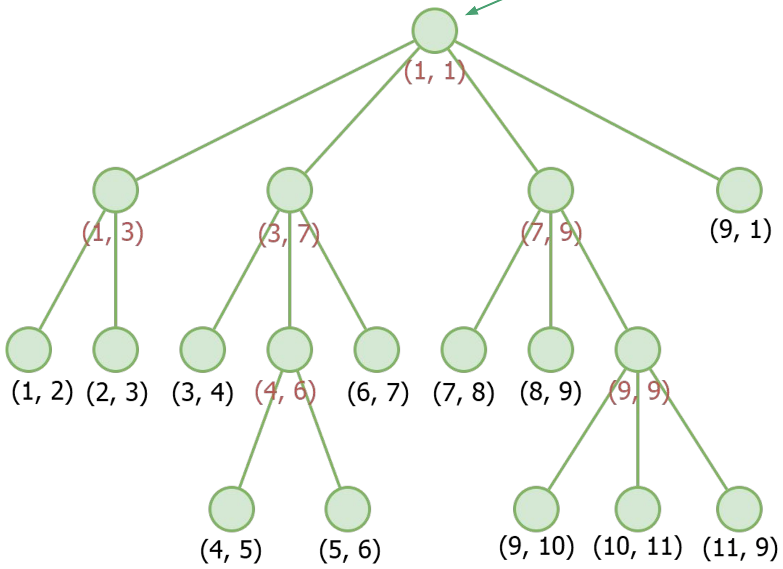
Budowa drzewa dla grafu z punktami artykulacji





Analiza drzewa

Wywołujemy procedurę **table(v)** dla korzenia drzewa.



- Jeśli *v* jest liściem drzewa, zwracana jest tablica z domyślnymi wartościami.
- Jeśli *v* nie jest liściem drzewa, tablice jego potomków są łączone procedurą **merge**, a następnie naprawiane procedurą **adjust**

Wyznaczanie tablic

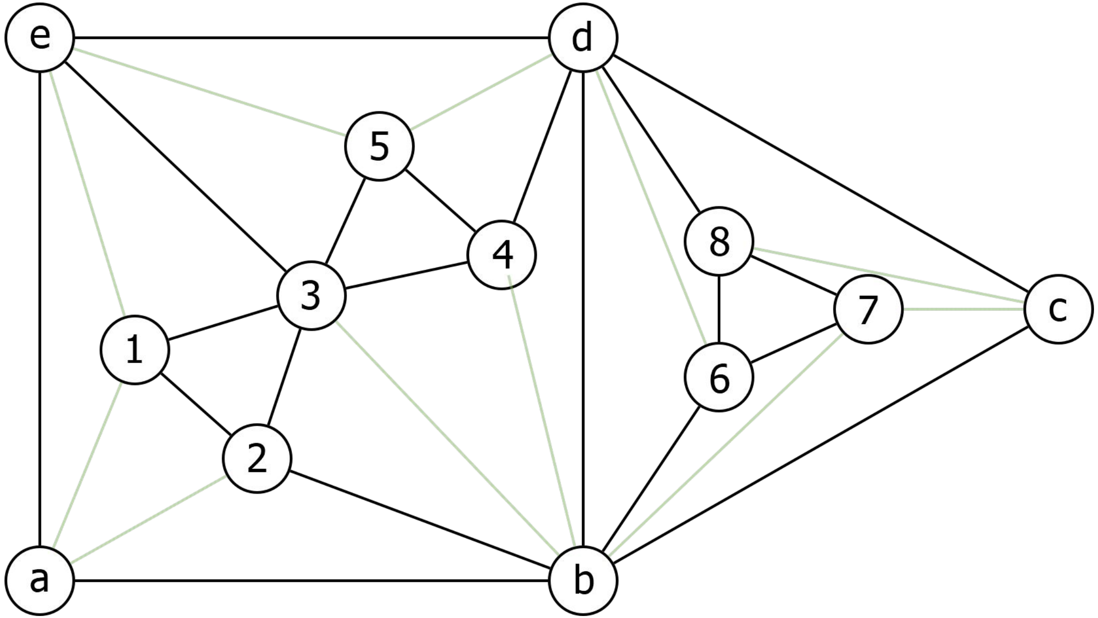
- Tablice w których zapisane są rozmiary największego zbioru niezależnego dla podgrafów wyznaczonych przez podgrafy drzewa
- Tablica dla jednej krawędzi (x, y):
  - 0 jak oba wierzchołki nie są w zbiorze
  - 1 jak x jest w zbiorze, ale nie y
  - 1 jak y jest w zbiorze, ale nie x
  - Nieokreślone jak oba są w zbiorze
- Wartości tablic wierzchołków innych niż liście liczona rekurencyjnie
- Kolejność łączenia tablic od lewa do prawa

Operacje na tablicach

- merge
  - Łączy dwie tablice T_1 i T_2 dla wierzchołków (x, y) oraz (y, z)
  - Powstaje tablica dla wierzchołka (x, z)
  - Dla każdej kombinacji elementów **x** ze T_1 i **z** z T_2 szuka takiego przypisania wspólnego wierzchołka **y**, żeby $T_1(x, y) + T_2(y, z) - |y|_1$ było największe
- adjust
  - Usunięcie niemożliwych przypadków z tablic

Algorytm dla grafu k-zewnętrznie planarnego

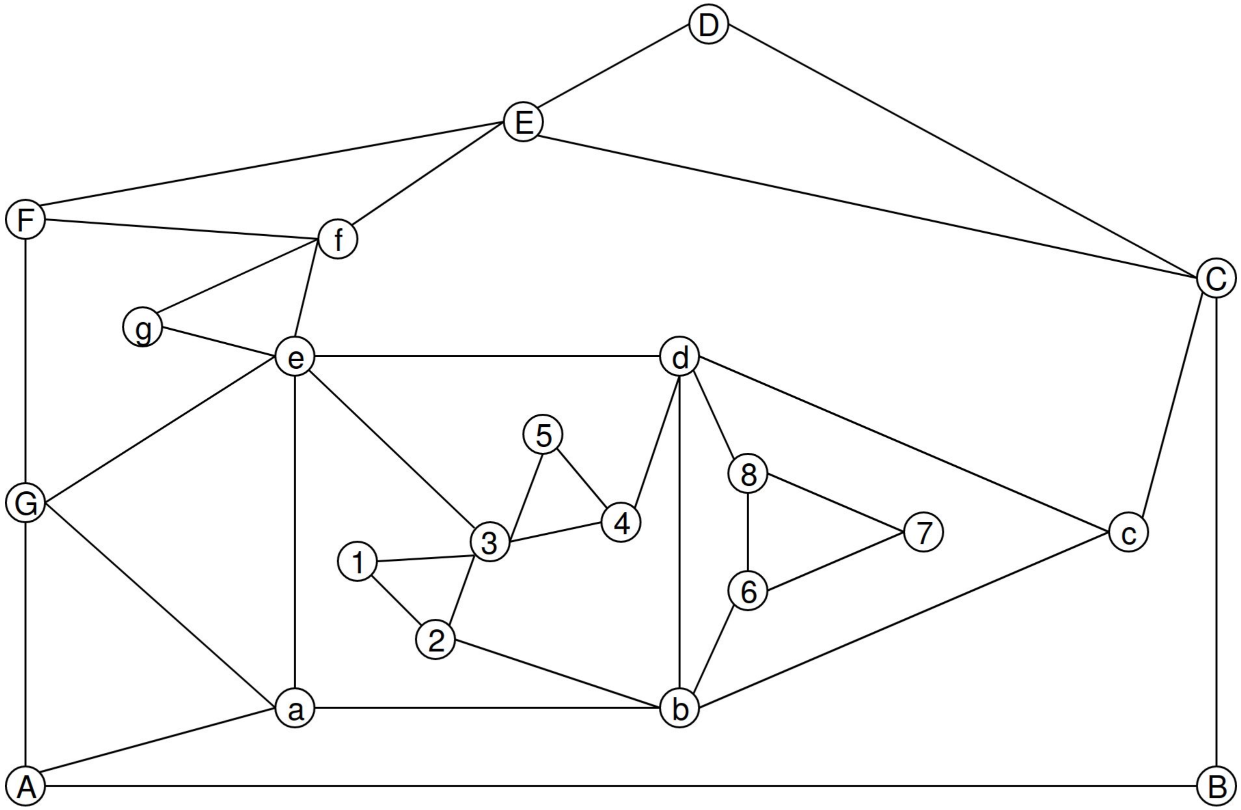
- Przetwarzanie wstępne (pojęcie **komponentu**)
- Tworzenie drzew



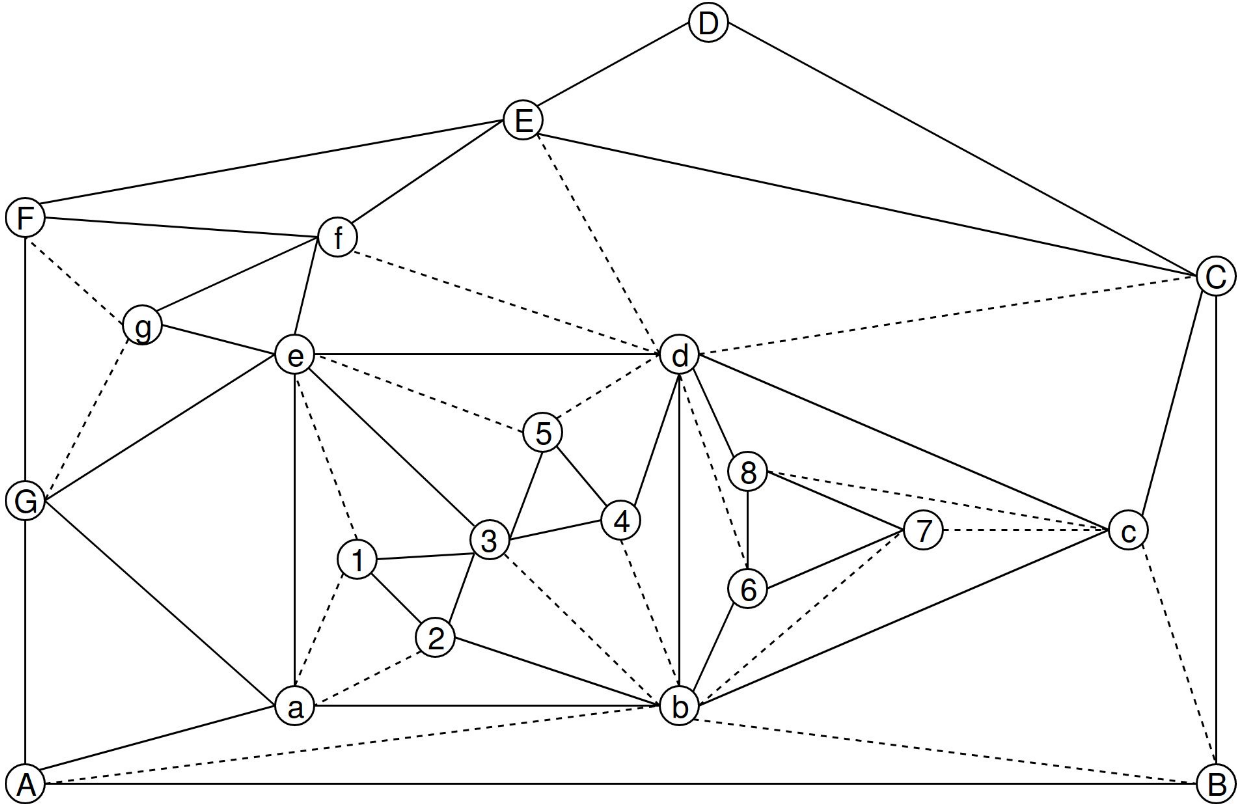
Algorytm dla grafu k-zewnętrznie planarnego

- Programowanie dynamiczne
- Podział grafu na podgrafy o wspólnych wierzchołkach brzegowych
- Graf planarny: ścieżka między wierzchołkami zewnętrznymi dzieli graf na dwie części
- Jak najkrótsze brzegi: maksymalnie 2k wierzchołków
- Kawałki wycinane za pomocą lewego i prawego brzegu
- Brzegi definiowane dla każdego wierzchołka drzew

Przykładowy graf



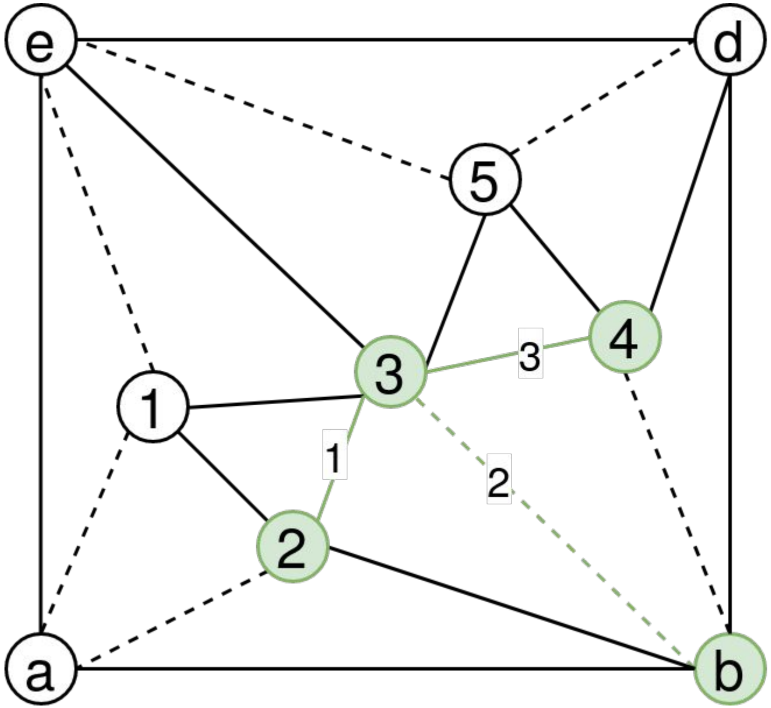
Graf po triangulacji

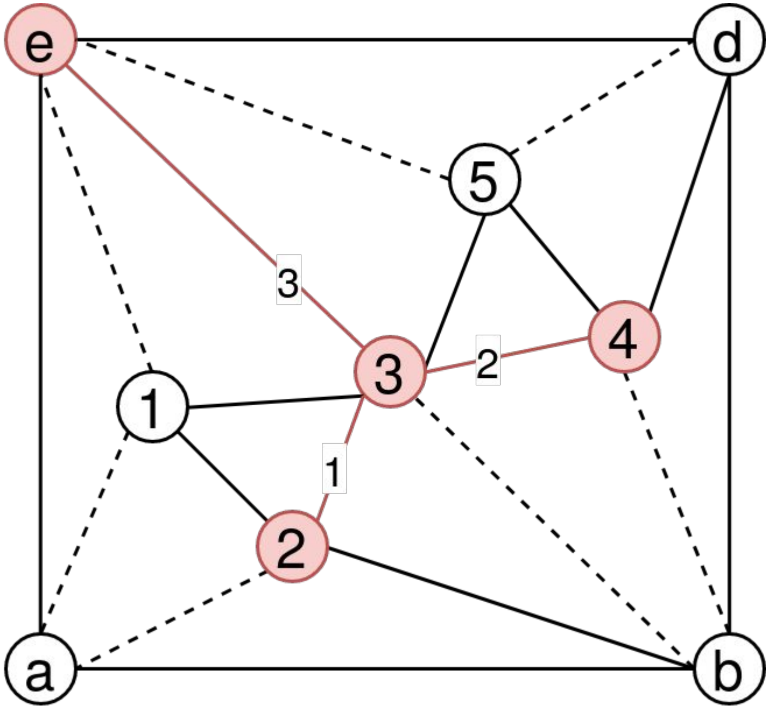


Jak wyznaczyć brzegi?

- Funkcje LB i RB, zwracają kolejny wierzchołek w brzegu
- Argumentem jest wierzchołek drzewa, nie grafu
- Wartość jest wybierana z etykiet dzieci wierzchołka regionu otaczającego aktualny komponent
- Punkt podziału (dividing point):
    - Definiowany dla dwóch kolejnych krawędzi (a, b) i (b, c) występujących w spacerze wokół komponentu C przeciwnie do wskazówek zegara
    - Wierzchołek z regionu otaczającego komponent
    - Wybieramy taki wierzchołek x sąsiadujący z b po triangulacji, że krawędzie (b, a), (b, x), (b, c) występują w kolejności przeciwnej do wskazówek zegara wokół b

Punkt podziału dla krawędzi (2, 3) i (3, 4)





e nie może być punktem podziału, bo krawędzie występują wtedy w złej kolejności

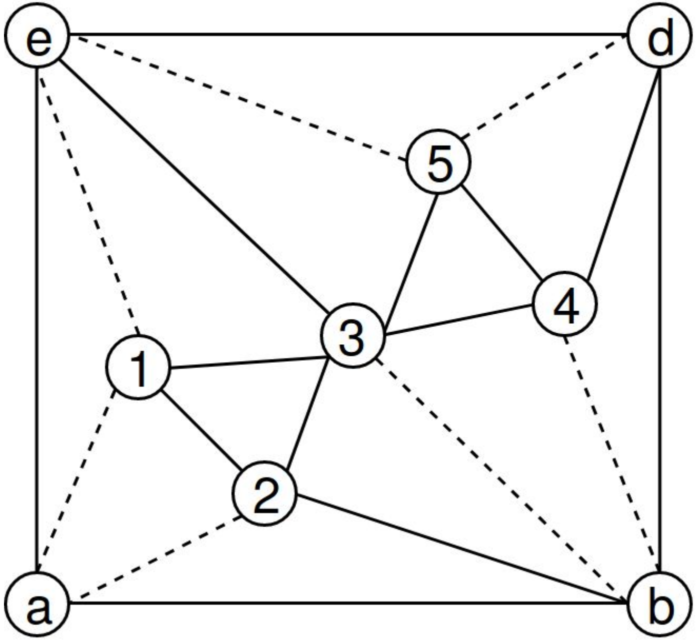
Definicja LB i RB dla krawędzi

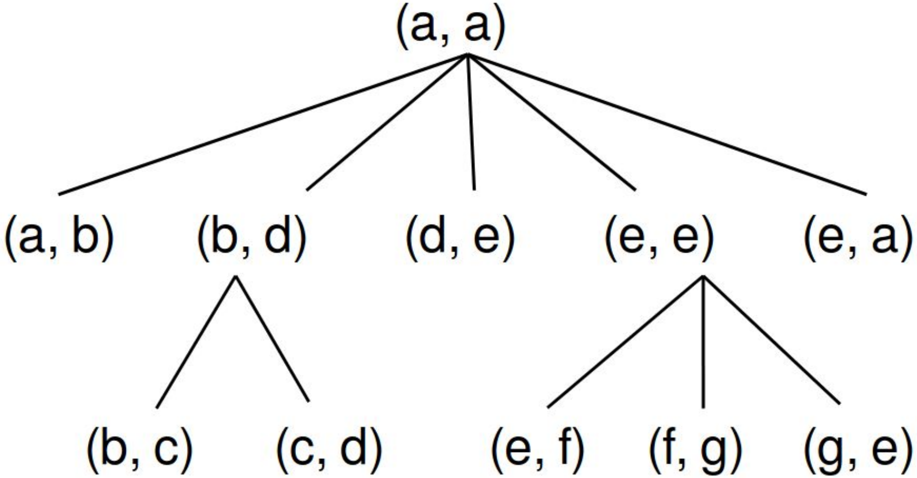
- Lewy brzeg pierwszego liścia: 1
- Prawy brzeg ostatniego liścia: r + 1, r to liczba dzieci regionu otaczającego
- Wartości LB niemalejące
- LB dla pozostałych: pierwszy możliwy punkt podziału dla dwóch kolejnych liści
- Nakładanie się brzegów: RB liścia to LB następnego

Definicja LB i RB dla regionu

- LB dla regionu: LB pierwszego dziecka
- RB dla regionu: RB ostatniego dziecka

LB i RB - przykład dla komponentu 1 - 5

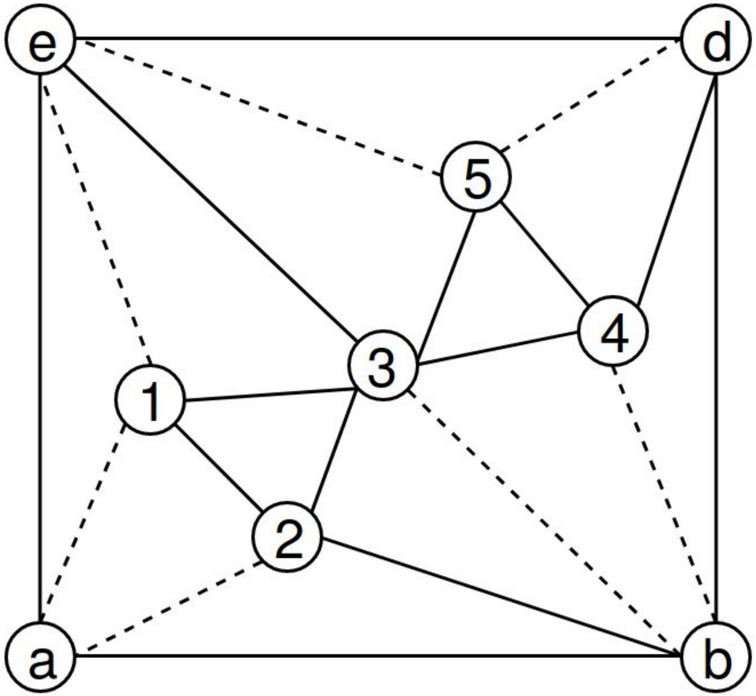


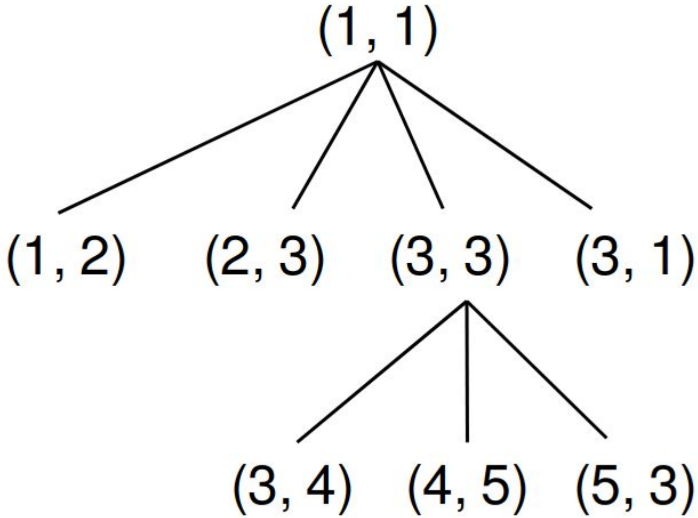


Etykiety dzieci f: a, b, d, e

Komponent jest otoczony regionem, który w drzewie jest wierzchołkiem (a, a)

LB i RB - przykład dla komponentu 1 - 5

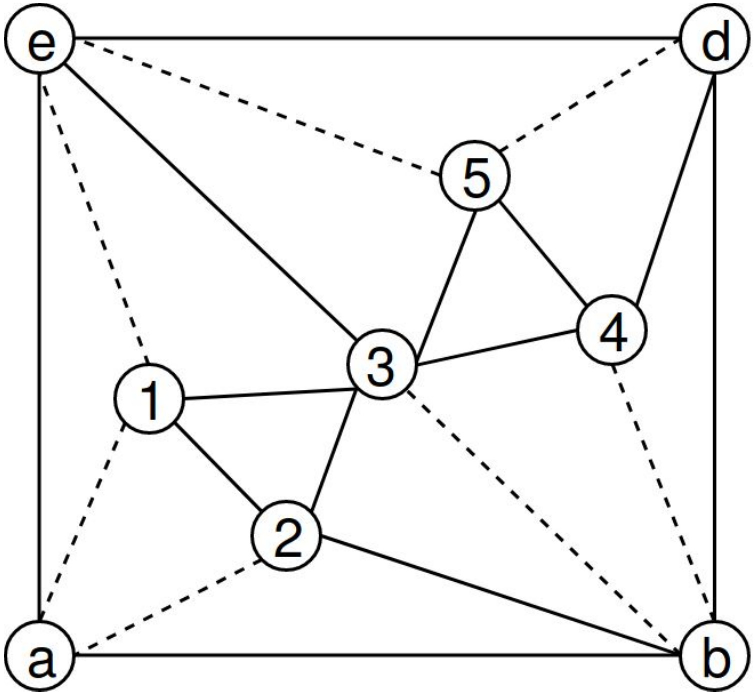


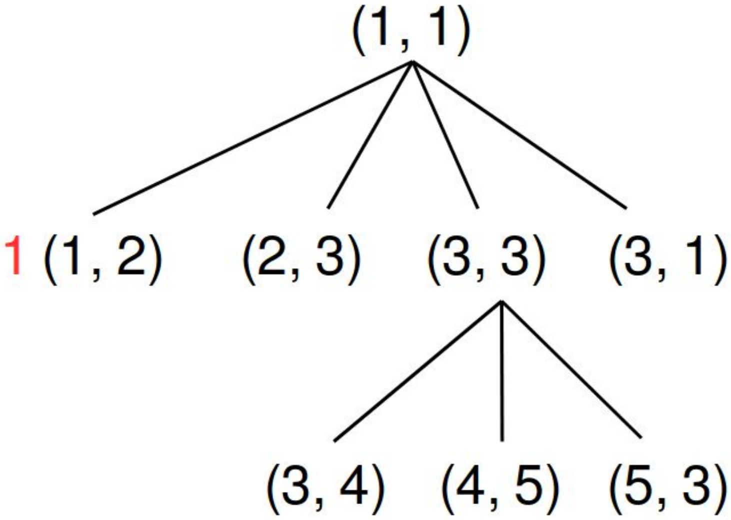


Etykiety dzieci f: a, b, d, e

Wartości LB i RB wyznaczane przy przechodzeniu drzewa post order

LB i RB - przykład dla komponentu 1 - 5

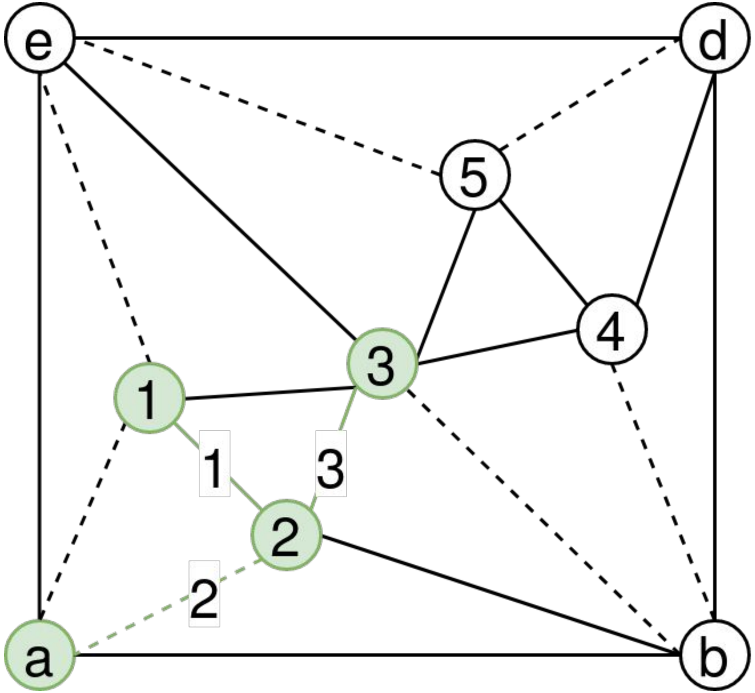


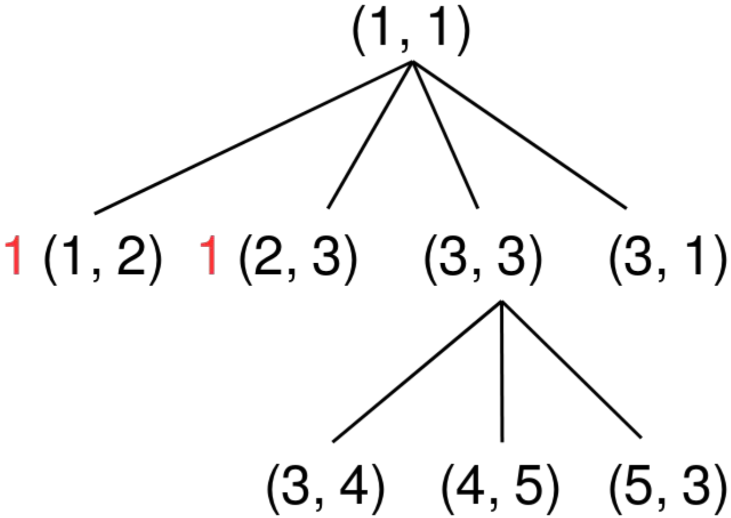


Etykiety dzieci f: a, b, d, e

Lewy brzeg pierwszego liścia to 1

LB i RB - przykład dla komponentu 1 - 5

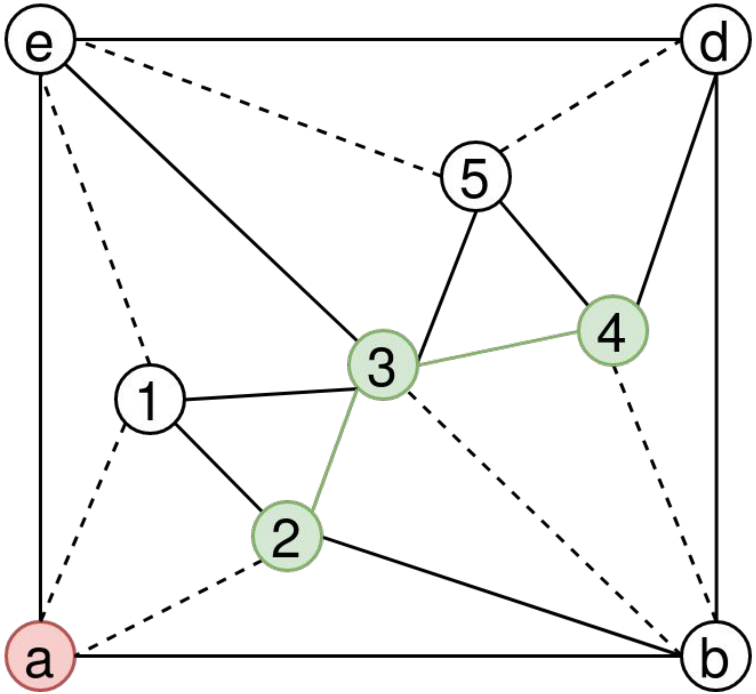


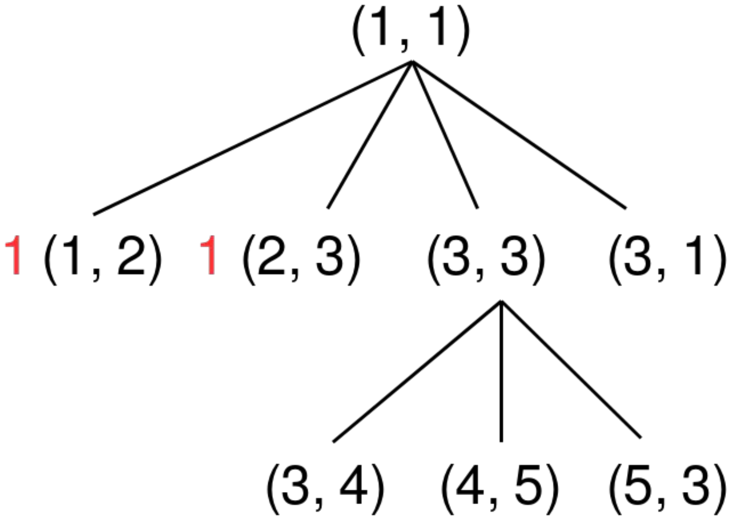


Etykiety dzieci f: a, b, d, e

a jest punktem podziału dla (1, 2) i (2, 3)

LB i RB - przykład dla komponentu 1 - 5

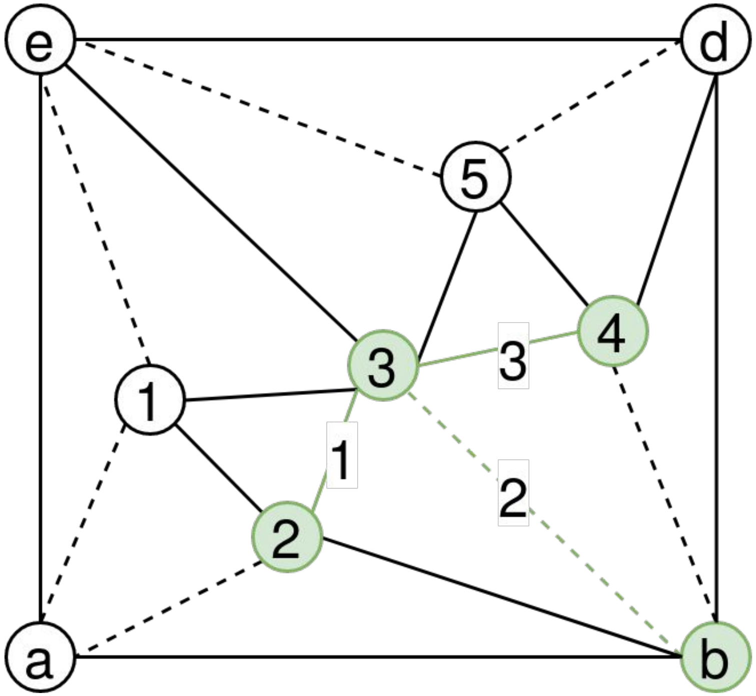


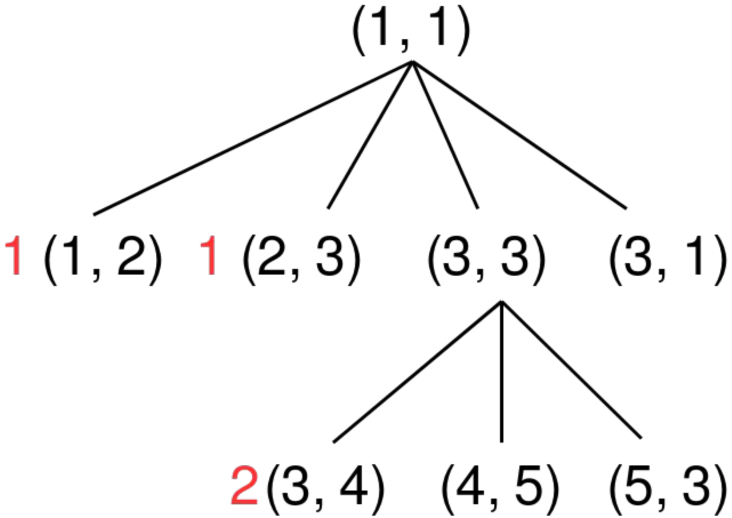


Etykiety dzieci f: ~~a~~, b, d, e

a nie jest punktem podziału dla (2, 3) i (3, 4), bo nie sąsiaduje z 3

LB i RB - przykład dla komponentu 1 - 5

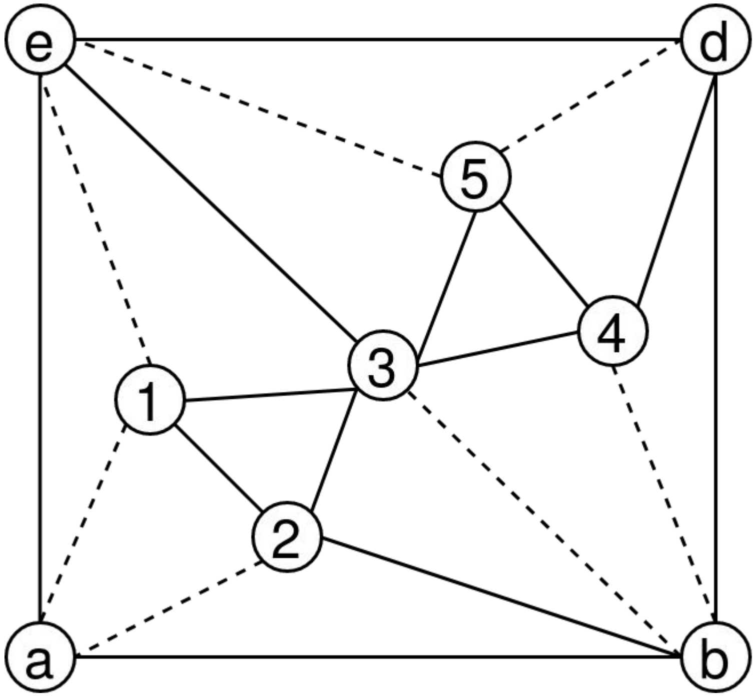


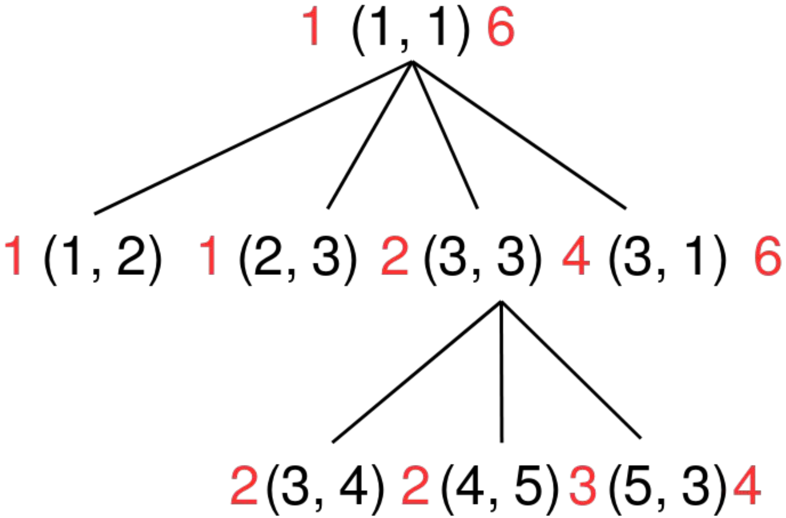


Etykiety dzieci f: ~~a~~, b, d, e

b jest punktem podziału dla (2, 3) i (3, 4)

LB i RB - przykład dla komponentu 1 - 5



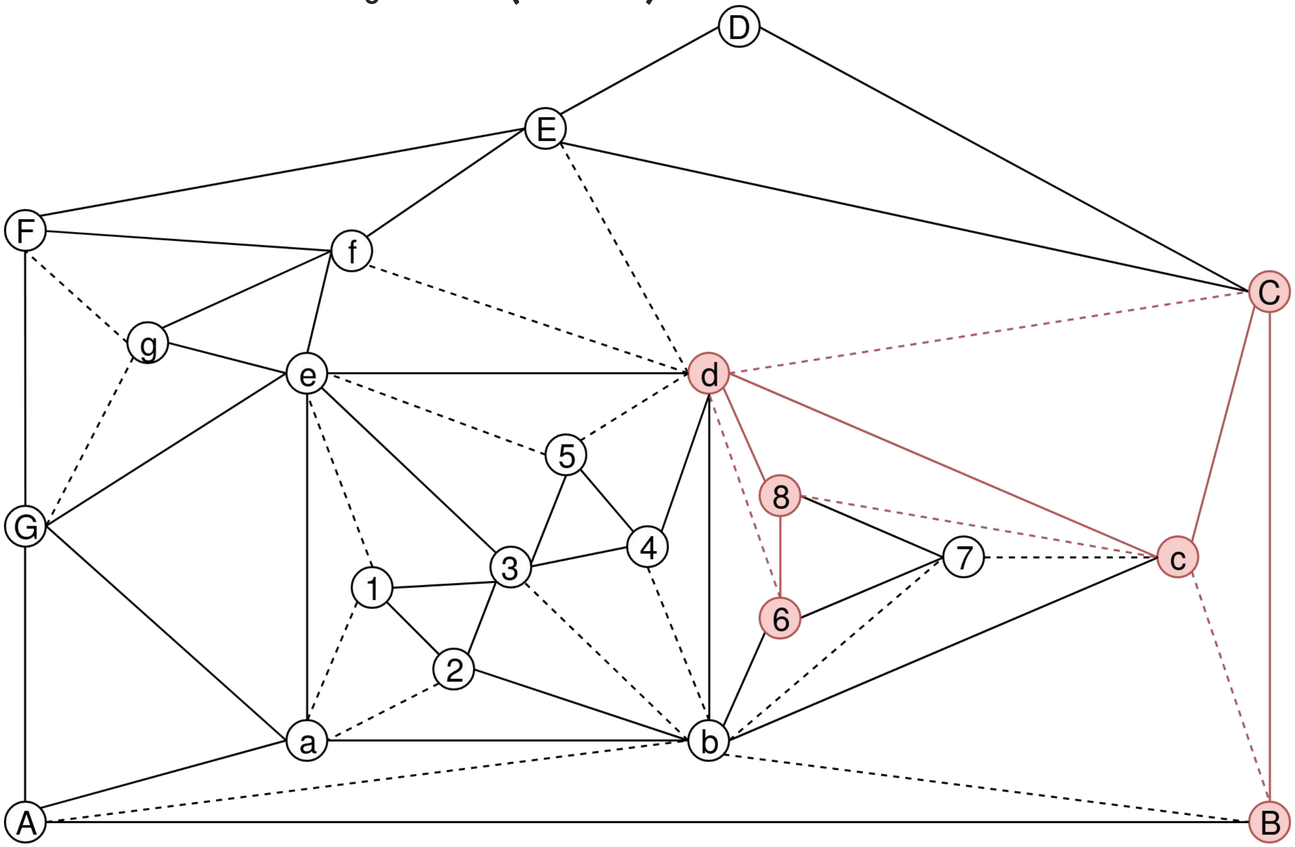


Reszta drzewa wypełniona stosując reguły dla liści i krawędzi

Definicja kawałka (slice)

- Region z etykietą (x, y) nie zawierający w sobie wierzchołków: (x, y) + suma kawałków dzieci w drzewie
- Region z etykietą (x, y), który zawiera komponent C: (x, y) + kawałek dla korzenia drzewa C
- Krawędź zewnętrzna (x, y): kawałkiem jest ta sama krawędź
- v to krawędź wewnętrzna w regionie f, jego j-te dziecko ma etykietę (u_j, u_j+1)
  - LB(v) = RB(v) = j: (x, y) + (x, u_j) + (y, u_j) + lewy brzeg j-tego dziecka f
  - Wpp. dla j = LB(v), ..., RB(v): (x, y) + (x, u_j) + (y, u_j). Jeśli j < RB(v) to też dodajemy kawałek j-tego dziecka f

Kawałek dla krawędzi (8, 6)



Wyznaczanie tablic

- Tablice w których zapisane są rozmiary największego zbioru niezależnego dla różnych konfiguracji wierzchołków brzegowych
- Tablica dla jednej krawędzi (x, y):
  - 0 jak oba wierzchołki nie są w zbiorze
  - 1 jak x jest w zbiorze, ale nie y
  - 1 jak y jest w zbiorze, ale nie x
  - Nieokreślone jak oba są w zbiorze
- Tablice dla kawałków wyznaczane przy wyznaczaniu kawałka
- Kolejność łączenia tabeli dana przez definicję kawałków

Operacje na tablicach

- merge
  - Łączy dwie tablice T_1 i T_2 dla kawałków o wspólnym brzegu
  - Powstaje tablica dla kawałka, którego lewy brzeg jest taki sam jak lewy brzeg T_1, a prawy brzeg taki sam jak prawy brzeg T_2
  - Dla każdej kombinacji elementów x ze T_1 i z z T_2 szuka takiego przypisania wspólnego brzegu y, żeby $T_1(x, y) + T_2(y, z) - |y|_1$ było największe
  - Złożoność: O(2^k · 2^k · 2^k) = O(8^k)
- adjust
  - Usunięcie niemożliwych przypadków z tablic
  - Złożoność: O(2^2^i) = O(4^i)

Operacje na tablicach - c.d.

- contract
  - Zamienia tablicę na tablicę z poprzedniej warstwy
  - Połowa mniej elementów
  - Używane gdy jest region otaczający komponent
  - Korzeń drzewa komponentu: (z, z), porównujemy przypisania gdzie z jest w zbiorze i te gdzie nie jest
  - Złożoność: O(4^k)

Operacje na tablicach - c.d.

- create
  - Dwa parametry:
    - v - liść z drzewa dla którego tworzy się tablicę, ma etykietę (x, y)
    - p - liczba naturalna nie większa niż t + 1, t - liczba dzieci regionu otaczającego
  - Oblicza tabelę dla podgrafu
    - Zawierającego (x, y)
    - Dla p < t + 1: również prawy brzeg p-tego dziecka regionu otaczającego
    - Dla p = t + 1: również lewy brzeg p-tego dziecka otaczającego
    - Wszystkie krawędzie od x lub y do sąsiedniej warstwy wyżej wybranego brzegu
  - Złożoność: O(k4^k)

Operacje na tablicach - c.d.

- extend
  - Rozszerza tabelę dla komponentu z poziomu i o wierzchołek z warstwy i + 1
  - Podwaja rozmiar tablicy
  - Złożoność: O(4^k)

Procedura wyznaczania tablic

- Funkcja table
- Argument: v - element drzewa z etykietą (x, y)
- Wynik: Tablica dla v
- Wywoływane dla korzenia komponentu zewnętrznego
- Wynikiem będzie maksimum z otrzymanej tablicy
- Podzielona na cztery przypadki, takie same jak w definicji kawałków

Procedura wyznaczania tablic - c.d.

1. v reprezentuje region, który nie zawiera w sobie dalszych komponentów

```
u = v.children[0] #u to pierwsze dziecko v od lewej
T = table(u)
for c in v.children[1 : ]: #dla każdego kolejnego dziecka v
    T = merge(T, table(c))
return adjust(T)
```

2. v reprezentuje region, który zawiera komponent C

```
TC = tree(C) #drzewo dla komponentu C
return adjust(contract(table(TC.root)))
```

3. v reprezentuje krawędź zewnętrzną - zwracana tablica dla krawędzi (x, y)

Procedura wyznaczania tablic - c.d.

4. v reprezentuje krawędź (x, y) z warstwy i > 1

```
f = region otaczający komponent z v
vf = vertex(f) #wierzchołek w drzewie reprezentujący f
niech dzieci vf będą miały etykiety (z1, z2), (z2, z3), ..., (zm, z(m+1))
# Wybieramy indeksy wierzchołków z etykiet dzieci vf wszystkie te,
# dla których jest krawędź do y i których indeks należy do przedziału [LB(v), RB(v)]
adjecant_z = [i for i in range(LB(v), RB(v) + 1) if adjecant(y, zi)]
if len(adjecant_z) > 0:
    # Istnieją takie indeksy
    p = min(adjecant_z) #wybieramy najmniejszy indeks
else:
    p = RB(v)
T = create(v, p)
j = p - 1
while j >= LB(v):
    T = merge(extend(x, table(vf.children[j])), T)
    j = j - 1
j = p
while(j < RB(v)):
    T = merge(T, extend(y, table(vf.children[j])))
    j = j + 1
return T
```

Złożoność wyznaczania tablic

- adjust, contract i create wywoływane maks. raz w każdym wywołaniu table
- extend i merge wywołujemy nie częściej niż table
- Wniosek: merge jest operacją dominującą, ma największą złożoność i jest najczęściej wywoływane
- table jest wywoływane dokładnie raz dla każdego wierzchołka drzew
- Liczba wierzchołków drzew = liczba krawędzi w grafie + liczba regionów
- W grafie planarnym:
  - Liczba krawędzi $\leq 3n - 6$
  - Liczba regionów $\leq 2n - 4$
- Złożoność: O(n8^k)

Złożoność i jakość algorytmu

- Złożoność:
  - Jedna iteracja: $O(n1\ 8^k + n2\ 8^k + \ldots) = O(n\ 8^k)$
  - Powtarzamy k+1 razy: $O(kn8^k)$
- Jakość:
  - Usuwamy warstwy przystające modulo k + 1 do i
  - Musi istnieć takie i, że w warstwach usuniętych będzie maksymalnie 1 / (k + 1) wierzchołków z optymalnego rozwiązania
  - Dla komponentów mamy dokładny algorytm
  - Ostateczny rozmiar: nie mniejszy niż k / (k + 1) optymalnego

Źródła

- Brenda S. Baker. “Approximation Algorithms for NP-complete Problems on Planar Graphs”. In: J.ACM41.1 (Jan. 1994), pp. 153–180. ISSN: 0004-5411.doi:10.1145/174644.174650.
- Whittington, Philip. "Baker’s approximation scheme for planar graphs."
- Hajiaghayi, Mohammad, and Erik Demaine. "Approximation Schemes for Planar Graph Problems." Encyclopedia of Algorithms (2014): 1-5.
- Lipton, R. J., and Tarjan, R. E. 1979. A separator theorem for planar graphs. SIAM J. Appl. Math. 36, 2, 177-189.