Aproksymacja

Andrzej Lamecki
Michał Możdżonek
Jan Mycka

Agenda

- (nie)śmieszna historyjka z aproksymacją w tle
- trochę przydatnych definicji
- PTAS na podstawie algorytmu dokładnego
  - “Ale o co chodzi?”
  - Strukturyzacja wejścia
  - Strukturyzacja wyjścia
- metoda “Brandy Baker” - znajdowanie maksymalnego zbioru niezależnego w grafie planarnym

Aproksymacja - po co?

- Najciekawsze problemy = najtrudniejsze problemy
- Ograniczenia czasowe
- Problemy optymalizacyjne
  - minimalny pokrywający zbiór wierzchołków
  - maksymalny zbiór niezależny w grafie

Definicja

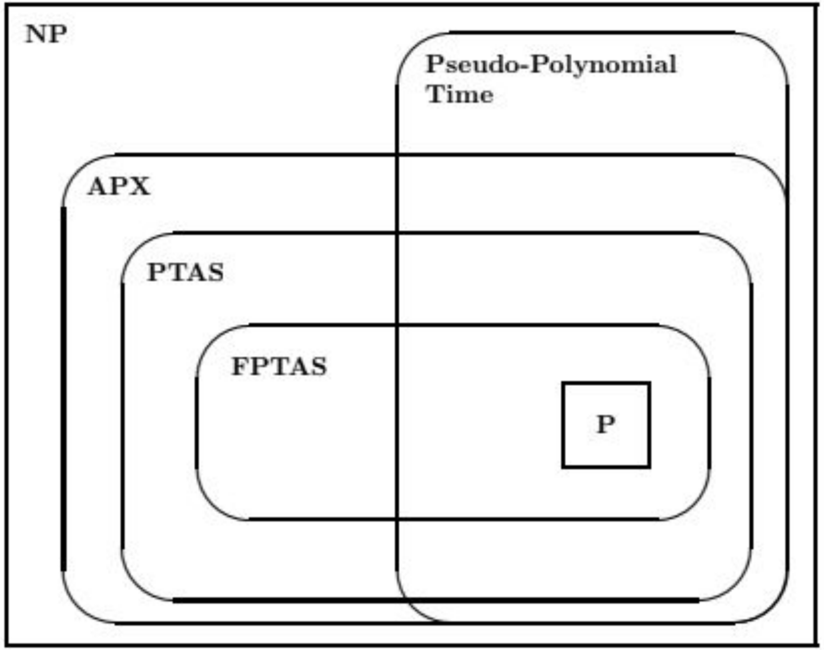
Niech X będzie problemem minimalizacyjnym (odpowiednio maksymalizacyjnym). Niech $\varepsilon > 0$ i niech $\rho = 1 + \varepsilon$ (odpowiednio $\rho = 1 - \varepsilon$). Algorytm A jest nazywany ρ-aproksymacyjnym dla problemu X, jeśli dla wszystkich instancji I problemu X wartość $| A(I) - OPT(I) |$ jest nie większa niż $\varepsilon * OPT(I)$.

Techniki projektowania

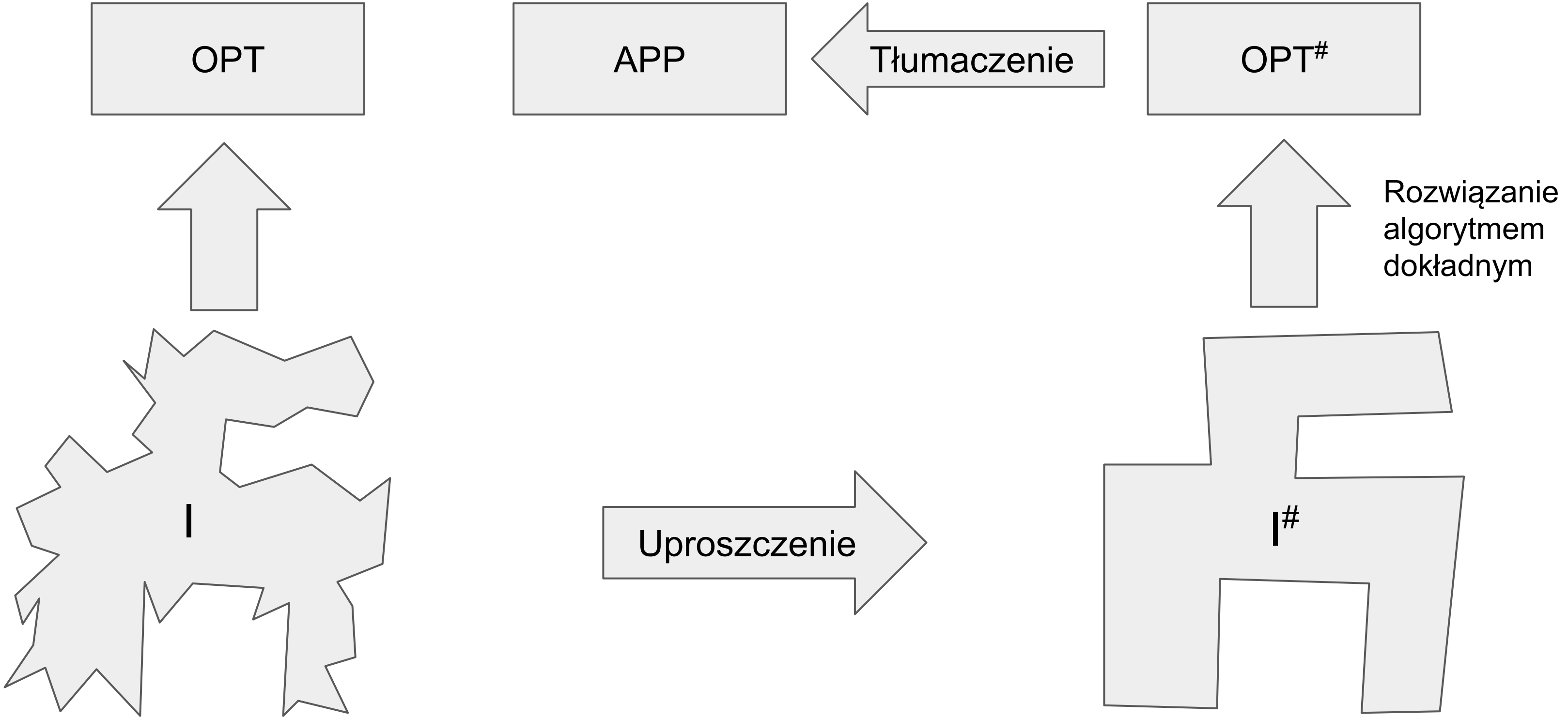
- Algorytm zachłanny
- Przeszukiwanie lokalne
- Programowanie dynamiczne
- Optymalizacja wypukła
- Losowe próbkowanie

Podział klas problemów

- APX - approximable
- PTAS - polynomial time approximation scheme
- FPTAS - full polynomial time approximation scheme



Strukturyzacja wejścia



OPT
APP
Tłumaczenie
OPT#
Rozwiązanie
algorytmem
dokładnym
I
Uproszczenie
I#

Kilka podstawowych technik

- Zaokrąglanie
- Łączenie
- Odcinanie
- Wyrównywanie

Problem podziału zadań na dwie identyczne maszyny

Wejście:
n zadań J_j $(j = 1, ..., n)$ o czasie wykonywania p_j

Zadanie:
Rozdysponować zadania na dwie identyczne maszyny, tak aby zminimalizować czas działania maszyny, która zakończy działanie jako ostatnia.

Założenia:
- wszystkie zadania są dostępne w dowolnym momencie
- nie można dzielić wykonania zadania na mniejsze części

Przydatne oznaczenia

$$p_{sum} = \sum_{j=1}^{n} p_j$$

$$p_{max} = \max_{j=1 \ldots n} p_j$$

$$L = \max\left\{\frac{1}{2} p_{sum}, p_{max}\right\}$$

Dolne ograniczenie na OPT:

$$L \le OPT$$

Faza A) Uproszczenie

Podzielimy zadania na “małe” i “duże”

$\leq \varepsilon L$

$> \varepsilon L$

sklejanie

S

$\left\lfloor \frac{S}{\varepsilon L} \right\rfloor$ *bloków o długości* εL

Ile mamy zadań w $I^{\#}$?

Do $I^{\#}$ wrzucamy:

- duże zadania z I
- bloki powstałe w wyniku sklejania mały zadań

Teraz każde zadanie z $I^{\#}$ ma długość co najmniej eps L

$$p_{sum} \leq 2L$$

Więc liczba zadań nie przekracza $\frac{p_{sum}}{\varepsilon L} \leq \frac{2L}{\varepsilon L} = \frac{2}{\varepsilon}$

Faza B) - rozwiązanie $I^{\#}$

Mamy $\frac{2}{\varepsilon}$ zadań, co daje $2^{\frac{2}{\varepsilon}}$ możliwych rozwiązań.

Czas wykonania każdego rozwiązania możemy policzyć w czasie liniowym $O\left(\frac{2}{\varepsilon}\right)$

Ostatecznie rozwiązanie $I^{\#}$ znajdujemy w czasie $O\left(2^{\frac{2}{\varepsilon}} \cdot \frac{2}{\varepsilon}\right)$

Przez $OPT^{\#}$ oznaczmy optymalne rozwiązanie instancji problemu $I^{\#}$

Jak daleko odeszliśmy z OPT# od OPT ?

Weźmy rozwiązanie optymalne OPT,

S_i - czas wykonania małych zadań na maszynie M_i

Skonstruujmy rozwiązanie ograniczające OPT# z góry:

- duże zadania zostawmy tak samo jak w OPT
- małe zadania zastąpmy przez $\left\lceil \frac{S_i}{\varepsilon L} \right\rceil$ bloków o długości εL

$$\left\lceil \frac{S_1}{\varepsilon L} \right\rceil + \left\lceil \frac{S_2}{\varepsilon L} \right\rceil \geq \left\lfloor \frac{S_1}{\varepsilon L} + \frac{S_2}{\varepsilon L} \right\rfloor = \left\lfloor \frac{S}{\varepsilon L} \right\rfloor$$

O ile zwiększyliśmy obciążenie M_i ?

$$\left\lceil \frac{S_i}{\varepsilon L} \right\rceil \varepsilon L - S_i \leq \left(\frac{S_i}{\varepsilon L} + 1 \right) \varepsilon L - S_i \;=\; \varepsilon L$$

$$OPT^{\#} \leq OPT + \varepsilon L \leq (1 + \varepsilon) OPT$$

Faza C) - tłumaczenie

Niech $L_i^{\#}$ - czas wykonywania na maszynie M_i

$B_i^{\#}$ - czas wykonywania dużych zadań na M_i

$S_i^{\#}$ - czas wykonywania małych zadań na M_i

$$L_i^{\#} = S_i^{\#} + B_i^{\#}$$

$$S_1^{\#} + S_2^{\#} = \varepsilon L \left\lfloor \frac{S}{\varepsilon L} \right\rfloor > \varepsilon L \left(\frac{S}{\varepsilon L} - 1 \right) = S - \varepsilon L$$

Konstruujemy rozwiązanie dla I

- duże zadania przydzielamy tak jak w $OPT^{\#}$
- na M_1 rezerwujemy czas $S_1{}^{\#} + 2\varepsilon L$
- na M_2 rezerwujemy czas $S_2{}^{\#}$

Pakujemy małe zadania na M_1, aż któreś nie zmieści

Na M_1 mamy $S_1 \geq S_1{}^{\#} + \varepsilon L$

Zostało nam $S - S_1{}^{\#} - \varepsilon L < S_2{}^{\#}$ ale to się zmieści na M_2

Ile tracimy w stosunku do OPT ?

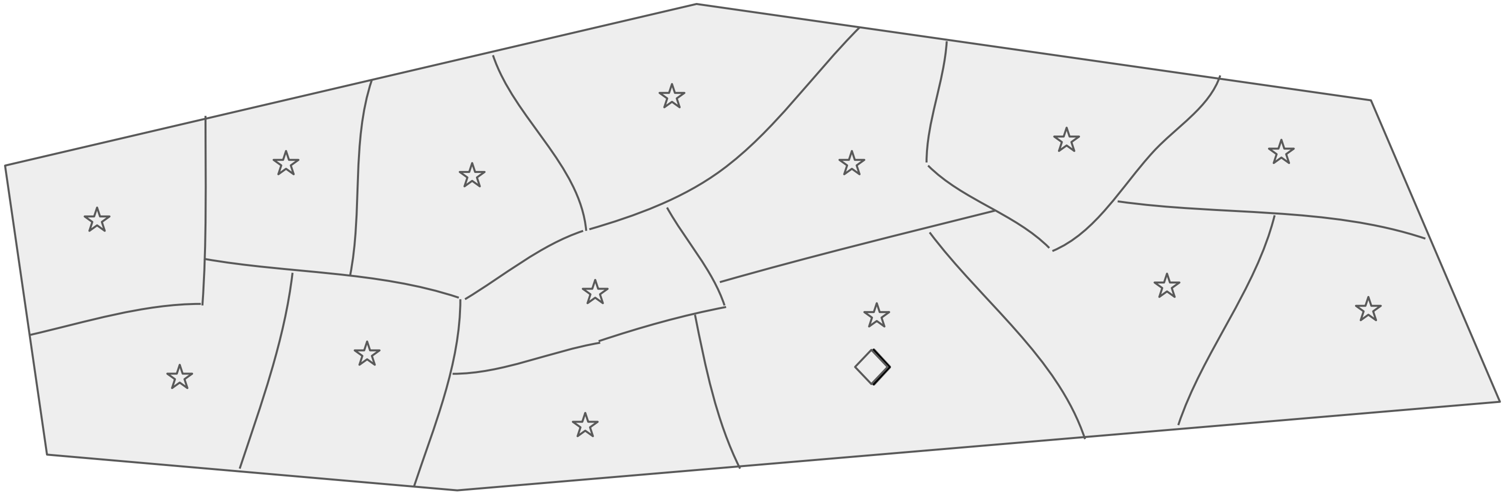
$$OPT \le {B_i}^{\#} + \left({S_i}^{\#} + 2\varepsilon L\right) = \ OPT^{\#} + 2\varepsilon L \le (1 + \ \varepsilon)OPT + 2\varepsilon L = \ (1 + 3\varepsilon)OPT$$

Złożoność: $O\left(n + \ 2^{\frac{2}{\varepsilon}} \cdot \frac{2}{\varepsilon} + n\right)$

Strukturyzacja wyjścia

Trzy kroki do zwycięstwa

A) Podział na regiony
B) Znalezienie reprezentantów
C) Wybranie najlepszego reprezentanta



Przydatne oznaczenia

$$p_{sum} = \sum_{j=1}^{n} p_j$$

$$L = \max\left\{\frac{1}{2}p_{sum}, p_{max}\right\}$$

$$p_{max} = \max_{j=1\ldots n} p_j$$

Dolne ograniczenie na OPT:

$$L \leq OPT$$

Faza A) - Podział

Dużych zadań mamy co najwyżej $\frac{2}{\varepsilon}$

F - zbiór wszystkich rozwiązań

Każde rozwiązanie $\sigma \in F$ określa przydzielenie n zadań do dwóch maszyn

Definicja regionów $F^{(l)}$

Rozwiązania σ_i σ_j leżą w tym samym regionie jeżeli σ_i przydziela każde duże zadanie do tej samej maszyny co σ_j

Warto zauważyć, że liczba regionów jest ograniczona przez $\frac{2}{\varepsilon}$

Faza B) -znalezienie reprezentantów

Rozważmy jeden region $F^{(k)}$

$B_i^{(k)}$ - czas wykonania dużych zadań na M_i

Na początek obciążenie na M_1, M_2 wynosi odpowiednio $B_1^{(k)}$, $B_2^{(k)}$

Dalej bierzemy kolejno małe zadania i przydzielamy je do maszyny z mniejszym obciążeniem.

Dostaliśmy $APP^{(k)}$

O ile $APP^{(k)}$ różni się od $OPT^{(k)}$?

Rozważmy maszynę M_i z większym obciążeniem

Ostatnie zadanie, które przydzieliliśy do M_i miało długość $\leq \varepsilon L$,
a obciążenie M_i wynosiło wtedy $\leq \frac{1}{2} p_{sum}$

$$APP^{(k)} \leq \frac{1}{2} p_{sum} + \varepsilon L \leq (1+\varepsilon)L \leq (1+\varepsilon)OPT \leq (1+\varepsilon)OPT^{(k)}$$

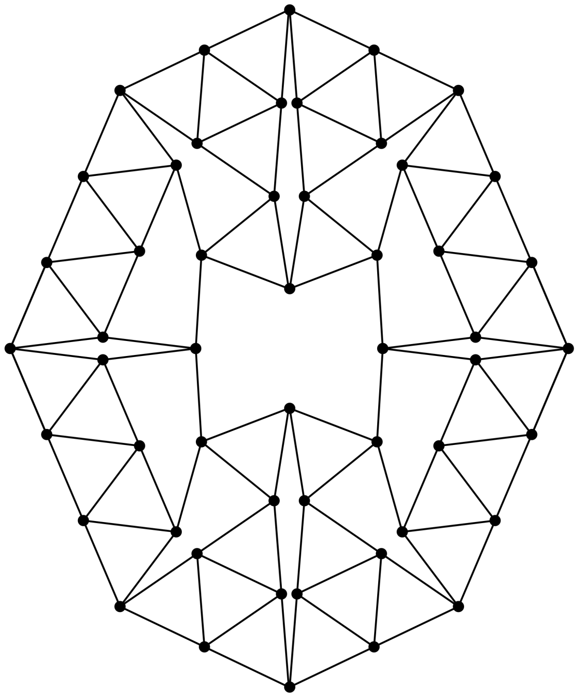
Faza C) wybranie najlepszego reprezentanta

Zostawiamy słuchaczowi w celu ćwiczenia.

Złożoność: $$O\left(2^{\frac{2}{\varepsilon}} \cdot n + \frac{2}{\varepsilon}\right)$$

Opis problemu

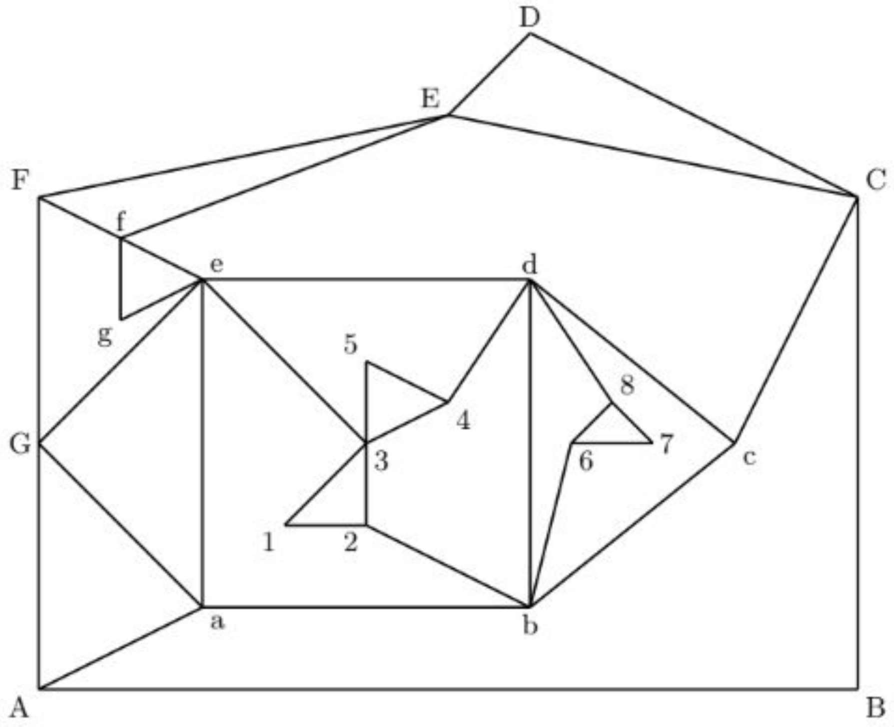
Znaleźć największy zbiór niezależny w grafie planarnym.



Definicje

- Grafem **zewnętrznie planarnym** (ang. **outerplanar**) nazywamy graf planarny, który posiada rysunek planarny, na którym wszystkie wierzchołki należą do ściany zewnętrznej
- Grafem **k-zewnętrznie planarnym** (ang. **k-outerplanar**) nazywamy graf planarny, który posiada rysunek planarny, na którym podgraf indukowany przez wszystkie wierzchołki nienależące do zewnętrznej ściany jest **(k-1)-zewnętrznie planarny**

Grafy k-zewnętrznie planarne



Graf 3-zewnętrznie planarny [Baker, 1994]

Metoda Brendy Baker

Metoda Brendy Baker

1. Znajdź zanurzenie planarne grafu i podziel go na warstwy
2. Dla i=0..k
    2.1. Usuń z grafu warstwy o numerze przystającym do i mod k
    2.2. Znajdź dokładne rozwiązanie dla powstałych składowych
    2.3. Zsumuj rozwiązania dla składowych
3. Zwróć maksymalny wynik spośród k obliczonych w punkcie 2.

Ile tracimy?

Niech

S - największy zbiór niezależny w G

L_i - podgraf G indukowany przez warstwy $i \ mod \ k$

$S_i = S \cap L_i$

$$\forall_{\substack{i,j=0..k \\ i \neq j}} S_i \cap S_j = \emptyset \qquad \bigcup_{i=0}^{k} s_i = s \qquad \exists_i |S_i| \leq \frac{|S|}{k}$$

Czy na pewno tylko tyle?

Niech

$$G_i = G \backslash L_i$$

S_i^* - największy zbiór niezależny w G_i

$$|S| - |S_i| \leq |S_i^*|$$

$$|S_i^*| \geq |S| - |S_i| \geq |S| - \frac{|S|}{k} \geq \frac{k-1}{k}|S|$$

Złożoność metody Baker

Tw.1 [Baker, 1994]

Niech G będzie grafem k-zewnętrznie planarnym. Znalezienie największego zbioru niezależnego w G jest możliwe w czasie $O(8^k n)$.

Złożoność metody

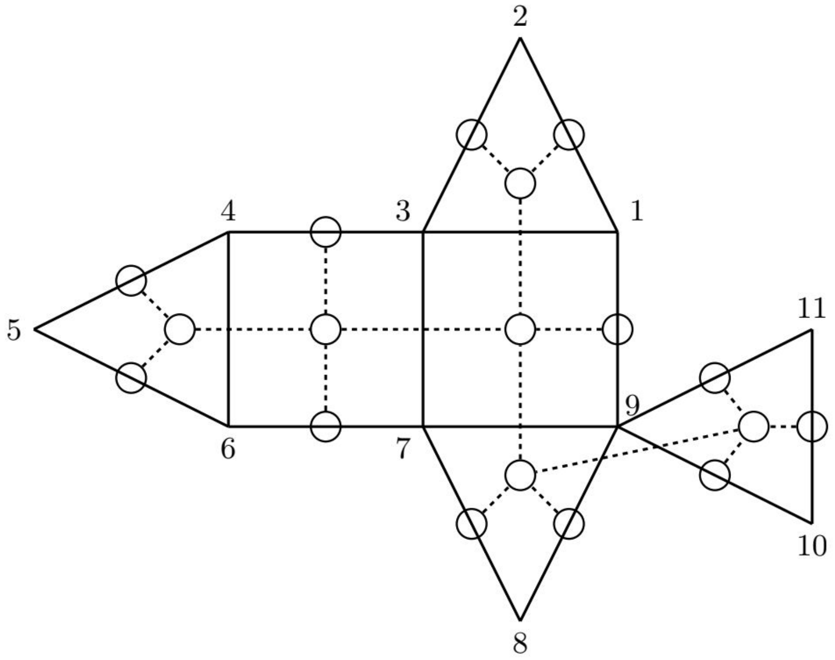
1. Znajdź zanurzenie planarne grafu i podziel go na warstwy
2. Dla i=0..k
    2.1. Usuń z grafu składowe przystające do i mod k
    2.2. Znajdź dokładne rozwiązanie dla powstałych składowych
    2.3. Zsumuj rozwiązania dla składowych
3. Zwróć maksymalny wynik spośród k obliczonych w punkcie 2.

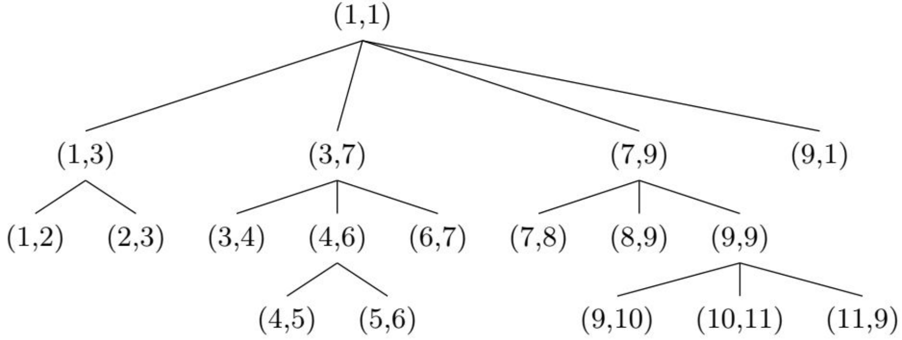
$O(8^k n)$

$$O(8^k kn)$$

Największy zbiór niezależny w grafie zewnętrznie planarnym

Konstrukcja drzewa przeszukiwania





Źródła

- “Approximation Schemes – A Tutorial” - Petra Schuurman, Gerhard J. Woeginger
- “Baker’s approximation scheme for planar graphs” - Philip Whittington