Uniwersytet Jagielloński w Krakowie

Wydział Matematyki i Informatyki

Pola Kyzioł

Nr albumu: 1092406

Tytuł pracy dyplomowej

Praca magisterska
na kierunku Informatyka Analityczna

Praca wykonana pod kierunkiem
dr hab. Tomasz Krawczyk
Instytut Informatyki Analitycznej

Kraków 2019

Oświadczenie autora pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

....	....
Kraków, dnia	Podpis autora pracy

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

....	....
Kraków, dnia	Podpis kierującego pracą

Spis treści

Rozdział 1

Dekompozycja drzewowa

1.1 Definicja dekompozycji i szerokości drzewowej

Dekompozycją drzewową grafu G nazywamy parę $\mathcal{T} = (T, \{X_t : t \in V(T)\})$, gdzie T jest drzewem, a $\{X_t : t \in V(T)\}$ zawiera zbiory wierzchołków grafu G i spełnia następujące warunki:

- Dla każdej krawędzi $\{u, v\} \in E(G)$, istnieje węzeł $t \in V(T)$, taki że $u \in X_t$ i $v \in X_t$.
- Dla każdego wierzchołka $v \in V(G)$, zbiór $\{t \in V(T) : v \in X_t\}$ jest poddrzewem drzewa T.

Od tej pory wierzchołki grafu wyjściowego G będą nazywane po prostu *wierzchołkami*, natomiast węzły drzewa T będą nazywane *kubełkami*.

Szerokość drzewowa dekompozycji drzewowej $\mathcal{T}$ jest zdefiniowana następująco:
$sd_{\mathcal{T}} = max_{t \in V(T)} |X_t - 1|$. Natomiast szerokość drzewowa grafu G jest minimalną szerokością drzewową wziętą po wszystkich możliwych dekompozycjach drzewowych G:
$sd_G = min\{sd_{\mathcal{T}} : \mathcal{T}$ jest dekompozycją drzewową $G\}$.

1.2 Ładna dekompozycja drzewowa

Dla uproszczenia posługiwania się dekompozycją drzewową przy definiowaniu algorytmów dynamicznych, będziemy używać tzw. *ładnej dekompozycji drzewowej*, która została po raz pierwszy wprowadzona przez Kloks [1].

Ładna dekompozycja drzewowa $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ musi spełniać następujące warunki:

- T jest ukorzenione.
- Każdy *kubełek* T ma co najwyżej dwoje dzieci.
- Jeśli *kubełek* t ma dwoje dzieci p i q, wtedy $X_t = X_p = X_q$.

- Jeśli *kubełek* t ma jedno dziecko p, to $|X_t| = |X_p|+1$ *oraz* $X_p \subset X_t$ albo $|X_t| = |X_p|-1$ *oraz* $X_t \subset X_p$.

Ponieważ w ładnej dekompozycji drzewowej, *kubełki* różnią się od siebie o co najwyżej jeden *wierzchołek*, każde przejście między jednym a drugim *kubełkiem* odpowiada dokładnie jednej operacji na grafie wyjściowym G. Każdy *kubełek* ma jeden z następujących pięciu typów:

- `WPROWADZAJĄCY v` - *kubełek* ten ma o jeden *wierzchołek* więcej niż jego jedyne dziecko: $X_p \cup \{v\} = X_t$. Każdy wierzchołek $v \in V(G)$, ma co najmniej jeden kubełek wprowadzający.

- `ZAPOMINAJĄCY v` - *kubełek* o jednym wierzchołku mniej niż jedgo jedyne dziecko: $X_t \cup \{v\} = X_p$. Jego specjalnym reprezentantem jest korzeń. Dla każdego wierzchołka $v \in V(G)$, istnieje dokładnie jeden kubełek zapominający.

- `SCALAJĄCY` - jedyny *kubełek* posiadający dwoje dzieci: $X_t = X_p = X_q$, scala dwa podgrafy o przecięciu X_t.

- `LIŚĆ` - dla t będącego liściem: $X_t = \emptyset$.

- `UZUPEŁNIAJĄCY uv` - *kubełek*, który nie pojawił się w pierwotnej definicji ładnej dekompozycji drzewowej, ale ułatwia definiowanie algorytmów operujących na dekompozycjach drzewowych. *Kubełek* uzupełniający wprowadza krawędź $uv \in E(G)$ (uzupełnia krawędziami reprezentację grafu G w drzewie T). *Kubełek* t `UZUPEŁNIAJĄCY` uv zawiera oba *wierzchołki* krawędzi: $u \in X_t$ i $v \in X_t$. Dla każdego uv istnieje dokładnie jeden kubełek uzupełniający i - przyjmując bez straty ogólności $t(u)$ jest przodkiem $t(v)$ (gdzie $t(v)$ to najwyższy *kubełek*, taki że $v \in X_{t(v)}$) - znajduje się on pomiędzy $t(v)$ a `ZAPOMINAJĄCY v`.

daj tu przykład

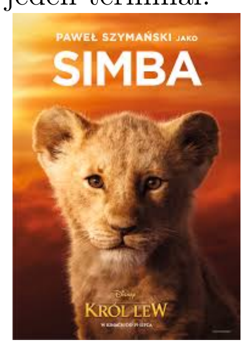
1.3 Obliczanie dekompozycji drzewowej

Rozdział 2

Klasyczne algorytmy dynamiczne

2.1 Drzewo Steinera

Mamy dany nieskierowany graf G oraz zbiór wierzchołków K będący podzbiorem $V(G)$, $K \subset V(G)$. Wierzchołki te nazywane są terminalami. Naszym zadaniem jest znalezienie dla grafu G takiego jego spójnego podgrafu H, który zawiera wszystkie terminale i jego rozmiar jest minimalny. Zakładamy, że mamy daną ładną dekompozycję drzewową grafu wyjściowego $G : \mathcal{T} = (T, \{X_t\}_{t \in V(T)})$. Dodatkowo, dla uproszczenia samego algorytmu, przyjmujemy, że każdy *kubełek* zawiera przynajmniej jeden terminal.



Bibliografia

[1] T. Kloks. *Treewidth. Computations and approximations.* Lecture Notes in Computer Science, 842, 1994.

[2] Albert Einstein. *Zur Elektrodynamik bewegter Körper*. (German) [*On the electrodynamics of moving bodies*]. Annalen der Physik, 322(10):891–921, 1905.

[3] Knuth: Computers and Typesetting,
`http://www-cs-faculty.stanford.edu/~uno/abcde.html`