1 Wprowadzenie

Na wykładzie poznaliśmy algorytm, który dla drzew ukorzenionych w czasie O(n) sprawdza czy nasze drzewo jest ukorzenione. Problem czy drzewa nieukorzenione sprowadzimy do tego samego problemu znajdując odpowiedni korzeń (wierzchołek centralny).

2 Algorytm

Najpierw znajdujemy możliwy korzeń (wierzchołek centralny) w drzewie T_1.

Indeksy wszystkich liście drzewa dodajemy na kolejkę Q_1, a następnie opróżniamy kolejkę usuwając te liście z drzewa i sąsiednie wierzchołki, które zostały liśćmi dodajemy do kolejki Q_2, i powtarzamy ten proces dopóki nie zostanie 1 lub 2 wierzchołki (więcej wierzchołków nie może być bo wtedy istnieje wierzchołek, który nie jest liściem, więc możemy jeszcze usunąć liście). Warto zanotować, że warunek pozostałych wierzchołków sprawdzamy za każdym razem przed usunięciem liści.

Podobnie wykonujemy dla T_2.

A następnie jeśli oba drzewa:

1. Mają po jednym możliwym korzeniu. Wtedy po prostu ukorzeniamy te drzewa w tych wierzchołkach i wykonujemy algorytm podany na wykładzie.
2. Mają po dwa możliwym korzeniu. Wtedy dwa razy uruchamiamy algorytm podany na wykładzie, dla pierwszego korzenia T_1 z pierwszym korzeniem T_2 oraz dla drugiego korzenia T_1 z pierwszym korzeniem T_2 (nie trzeba wykonywać sprawdzenia drugiego korzenia T_1 z drugim korzeniem T_2, ponieważ jest to analogiczne do sprawdzenia obu pierwszych korzeni).
3. Mają różną liczbę możliwych korzeni. Oznacza to, że mają różną liczbę wierzchołków centralnych więc nie mogą być izomorficzne.

3 Dowód poprawności

Wiemy o tym, że jeśli drzewa nie są izomorficzne to obojętnie, gdzie je ukorzenimy algorytm podany na wykładzie zwróci fałsz. Więc będziemy rozważać tylko drzewa izomorficzne T_1 i T_2 i sprawdzimy, czy dla nich otrzymaliśmy odpowiednie korzenie.

Fakt Z własności izomorfizmu wiemy, że jeśli wierzchołek w T_1 ma stopień 1 to jest on izomorficzny z innym wierzchołkiem stopnia 1 w T_2.

Czyli jeśli usuniemy wszystkie liście w T_1 i otrzymamy T_1' i podobnie dla T_2 otrzymamy T_2' to T_1' i T_2', będą one izomorficzne, ponieważ jeśli mieliśmy mieliśmy bijekcję f określającą, który wierzchołek T_1 jest izomorficzny z wierzchołkiem T_2 to obcieliśmy naszą funkcję dla wierzchołków z T_1', a f$[T_1']$= T_2'.

Korzystając z tej własności oczywiste jest, że wierzchołek centralny się zachowuje, więc pokażmy, że nasz algorytm znajduje wierzchołek centralny.

Teza: Algorytm znajduje wierzchołek centralny, N = promieć grafu.

Baza indukcji:

Dla N = 0, mamy jeden wierzchołek, więc na pewno jest naszym wierzchołkiem centralnym, co nam zwraca algorytm. Dla N = 1, mamy dwa wierzchołki, więc oba mogą być naszym wierzchołkiem centralnym, co nam zwraca algorytm. W przypadku trzech sprowadzamy to do N = 0.

Założenie indukcyjne: Dla dowolnego drzewa o promieniu N znajdujemy wierzchołek/i centralny/e.

Teza indukcyjna: Dla dowolnego drzewa o promieniu N+1 możemy znaleźć wierzchołek/i centralny/e.

Weźmy dowolne drzewo o promieniu N+1, jeśli usuniemy warstwe liśći wierzchołek centralny się źachowa", a my otrzymaliśmy drzewo o promieniu N i korzystając z założenia indukcyjnego znajdujemy wierzchołek centralny.

Więc na mocy zasady o indukcji algorytm znajduje wierzchołek centralny

4 Złożoność

Musimy usunąć n-1 lub n-2 wierzchołków, a do tego w przypadku przechowywania informacji tylko w tablicy będziemy musieli przejść tablicę sąsiadów dla każdego usuwanego wierzchołka, czyli każdą krawędź będziemy przechodzili maksymalnie 2 razy, więc nasza złożoność to O(n+m), ale ponieważ w drzewie liczba krawędzi wynosi n-1 to otrzymujemy O(n), dla algorytmu znajdowania wierzchołka centralnego, a sprawdzenie izomorfizmu ukorzenionych drzew też ma złożoność O(n), więc ostatecznie otrzymujemt O(n).

5 Usprawnienia

Zamiast tablicy moglibyśmy trzymać w wierzchołkach liste wskaźników na komórki w sąsiedzie, które wskazują na nas, dzięki temu nie musielibyśmy przechodzić wszystkich krawędzi 2 razy, a tylko raz.

Podczas szukania naszego wierzchołka centralnego moglibyśmy równocześnie szukać korzeń w T_1 i T_2 i przy usuwaniu „warstwy” liści sprawdzać ich liczebność.