Lista 4
Zadanie 3 nowej numeracji (2 starej)

Krystyna Korzonek

5 maja 2020

1 Algorytm

Ukorzeniamy drzewo. Dla każdego wierzchołka v T[v][0] będzie oznaczać maksymalną liczbę należących do zbioru niezależnego wierzchołków w poddrzewie v (w v i wszystkich jego potomkach), przy założeniu, że v nie należy do tego zbioru, a T[v][1] - przy założeniu, że należy. Tę tablicę wypełniamy funkcją licz. Następnie funkcją czytaj_wynik wypełniamy tablicę wynik, wynik[v] = 0 oznacza, że wierzchołek nie należy do maksymalnego zbioru niezależnego, 1 - że należy.

```
dla kazdego wierzcholka v
    T[v][0] = 0
    T[v][1] = 0
    wynik[v] = 0
licz(korzen) // korzen - dowolny wierzcholek drzewa
jesli T[korzen][1] > T[korzen][0]
    czytaj_wynik(korzen, true)
wpp
    czytaj_wynik(korzen, false)
return wynik


licz(v)
    jesli v jest lisciem
        T[v][1] = 1
```

```
    wpp.
        dla u - dzieci v
            licz(u)
        dla u - dzieci v
            T[v][0] += max(T[u][0], T[u][1])
            T[v][1] += T[u][0]
        T[v][1] += 1


czytaj_wynik(v, czy_wziety)
    jesli czy_wziety
        wynik[v] = 1
        dla u - dzieci v
            czytaj_wynik(czy_wziety, false)
    wpp.
        dla u - dzieci v
            jesli T[u][1] > T[u][0]
                czytaj_wynik(u, true)
            wpp
                czytaj_wynik(u, false)
```

2 Dowód poprawności

2.1 Moc zbioru niezależnego - poprawność tablicy T

Najpierw udowodnijmy, że wartości w tablicy T są obliczane poprawnie. To znaczy, że po wyjściu z funkcji licz(v) T[v][0] to maksymalna liczba należących do zbioru niezależnego wierzchołków w poddrzewie v przy założeniu, że v nie należy do tego zbioru, a T[v][1] - przy założeniu, że należy.
Udowodnijmy to indukcyjnie. Dla liści lemat w oczywisty sposób zachodzi. Załóżmy teraz, że zachodzi dla wszystkich dzieci wierzchołka v i pokażmy, że zachodzi dla v. Zauważmy, że w momencie w którym wyliczamy T[v], mamy już zapisane w tablicy poprawne wartości dla wszystkich dzieci v.
Jeśli v należy do zbioru niezależnego, to nie należy do niego żadne z jego dzieci. Suma wierzchołków należących ze zbioru niezależnego w poddrzewie v to 1 + suma liczb wierzchołków ze zbioru we wszystkich poddrzewach, których korzeniami są dzieci v, przy czym same dzieci nie mogą należeć do

zbioru. Maksymalne takie sumy są już poprawnie policzone w T[dziecko][0] (a oczywiście żeby z poddrzewie v wybrać najwięcej wierzchołków, w dzieciach też musimy wziąć najwięcej)
Jeśli v nie należy do zbioru niezależnego, to każde z jego dzieci możemy albo wziąć, albo nie wziąć. Wynik uzyskany na poddrzewie v będzie równy sumie wyników uzyskanych na jego dzieciach. Jeżeli w każdym dziecku będziemy brać maksymalny możliwy wynik, to w v też taki otrzymamy.

2.2 Należenie do zbioru niezależnego - poprawność tablicy wynik

Tablicę T wypełnialiśmy ”od liści”, tablicę wynik ”od korzenia”. Wobec tego i indukcję przeprowadzimy ”od korzenia”.
Pokażmy, że istnieje pewien największy zbiór niezależny taki, że dla każdego wierzchołka v wynik[v] = 1 dokładnie wtedy, kiedy v należy do tego zbioru.
Zauważmy najpierw, że wynik[v] = 1 wtedy i tylko wtedy, gdy uruchomimy funkcję czytaj_wynik z parametrem czy_wziety = true. W dodatku w każdym wierzchołku uruchamiamy tę funkcję dokładnię raz, a więc jeśli uruchomimy czytaj_wynik z czy_wziety = false, to na końcu algorytmu będzie zachodziło wynik[v] = 0.
Teza zachodzi dla korzenia. Funkcję czytaj_wynik uruchomimy w korzeniu z parametrem czy_wziety= true dokładnie wtedy, kiedy wzięcie korzenia pozwala stworzyć większy zbiór niezależny niż niewzięcie (jeśli w obu przypadkach moc zbioru niezależnego jest taka sama, po prostu ograniczamy się do takich zbiorów niezależnych, w których nie wybieramy korzenia, istnieje przynajmniej 1 taki zbiór).
Załóżmy teraz, że własność zachodzi dla pewnego wierzchołka v, i pokażmy, że zachodzi też dla jego syna u. Jeśli wybieramy v do zbioru niezależnego, to nie możemy wybrać u (a skoro v należy do zbioru niezależnego, to czytaj_wynik w v zostało uruchomione z true, a więc w u uruchomi się z false).
Jeśli nie wybieramy v to sytuacja jest analogiczna do opisanego wyżej wywoływania funkcji w korzeniu.

3 Złożoność

W funkcjach licz i czytaj_wynik w każdym wierzchołku wywołamy się maksymalnie raz, a więc złożoność jest liniowa.