Zadanie 3. *Zbiór $I \subseteq V$ zbioru wierzchołków w grafie $G = (V, E)$ nazywamy zbiorem niezależnym, jeśli żadne dwa wierzchołki z I nie są połączone krawędzią. Ułóż algorytm, który dla zadanego drzewa T znajduje najliczniejszy zbiór niezależny jego wierzchołków.*

Algorytm

Ukorzenimy drzewo w dowolnym wierzchołku. Pomysł stojący za algorytmem jest taki, że dla każdego wierzchołka v wyznaczymy najliczniejszy *zbiór niezależny* dla jego poddrzewa, który zawiera v, i taki, który v nie zawiera. Oczywiście jeden z nich będzie najliczniejszym zbiorem dla tego poddrzewa. Większy ze zbiorów wyznaczonych dla korzenia będzie najliczniejszym zbiorem niezależnym całego drzewa.

Przyjmiemy, że dla każdego wierzchołka będziemy wyznaczać parę zbiorów, gdzie pierwszy element tej pary to najliczniejszy zbiór niezależny poddrzewa zawierający ten wierzchołek, a drugi element to zbiór nie zawierający tego wierzchołka.

Dla dowolnego liścia v będzie to $(\{v\}, \varnothing)$. Dla pozostałych wierzchołków pierwszy zbiór będzie sumą drugich elementów dla jego potomków, a drugi zbiór będziemy tworzyć dodając większy ze zbiorów dla każdego z potomków.

```
Function MaxSet(A, B)
    if |A| ⩾ |B| then
        return A
    else
        return B
Procedure DFS(v, p)
    S_1 ← {v}
    S_2 ← ∅
    for u ∈ N(v) do
        if u ≠ p then
            (S'_1, S'_2) ← DFS(u, v)
            S_1 ← S_1 ∪ S'_2
            S_2 ← S_2 ∪ MaxSet(S'_1, S'_2)
    return (S_1, S_2)
Procedure LargestIndependentSet(T(V, E))
    v — wierzchołek z V, w którym ukorzeniamy drzewo
    (S_1, S_2) ← DFS(v, v)
    return MaxSet(S_1, S_2)
```

Poprawność

Lemat 1. *Dla dowolnego drzewa T i dowolnego wierzchołka v w tym drzewie, po ukorzenieniu T w v wyznaczona dla v para (S_1, S_2) to najliczniejsze zbiory niezależne T odpowiednio zawierające i nie zawierające v.*

Dowód. Indukcyjnie względem n — liczby wierzchołków w drzewie.

Podstawa indukcji. Dla $n = 1$ mamy pojedynczego liścia v, a $(\{v\}, \varnothing)$ jest poprawnie wyznaczoną parą zbiorów.

Krok indukcyjny. Niech $n > 1$. Załóżmy, że dla drzew o liczbie wierzchołków mniejszej niż n teza zachodzi. Niech T będzie dowolnym drzewem o n wierzchołkach. Niech v będzie dowolnym wierzchołkiem z T. Ukorzenimy T w v.

Niech $v_1, v_2, ..., v_k$ to synowie v. Poddrzewa w tych wierzchołkach mają mniej niż n wierzchołków. Zatem z założenia indukcyjnego, dla każdego z nich potrafimy wyznaczyć zbiory $(S_{i,1}, S_{i,2})$, które są największe możliwe.

Najpierw zajmiemy się największym zbiorem niezależnym zawierającym v. Taki zbiór nie może być utworzony z żadnego z $S_{1,1}, S_{2,1}, ..., S_{k,1}$, bo wszystkie zawierają bezpośrednich synów v, a więc zbiór nie byłby niezależny. Czy zatem zbiór $S_1 = \{v\} \cup S_{1,2} \cup S_{2,2} \cup ... \cup S_{k,2}$ jest optymalny? Każdy z rozłącznych zbiorów $S_{i,2}$ jest najliczniejszym zbiorem niezawierającym v_i dla danego poddrzewa. Dodatkowo te zbiory zostały wyznaczone dla niezależnych poddrzew, więc suma nadal jest niezależna. A więc tak wyznaczony S_1 jest optymalny.

Podobnie postępujemy, aby znaleźć największy zbiór niezależny niezawierający v. Teraz dla każdego niezależnego poddrzewa możemy wybrać większy ze zbiorów $S_{i,1}, S_{i,2}$, bo nie uwzględniając v, nie ryzykujemy naruszenia warunku niezależności. Stąd najliczniejszy zbiór niezawierający v to $S_2 = \textsc{MaxSet}(S_{1,1}, S_{1,2}) \cup ... \cup \textsc{MaxSet}(S_{k,1}, S_{k,2})$. □

Twierdzenie 1. *Algorytm znajduje najliczniejszy zbiór niezależny dla dowolnego drzewa T.*

Dowód. Z lematu 1 wiemy, że dla dowolnie wybranego v potrafimy wyznaczyć najliczniejsze zbiory niezależne T zawierające i niezawierające v. Oczywiście skoro najliczniejszy zbiór (już bez dodatkowych warunków) musi zawierać lub nie zawierać v, to jeden z wyznaczonych zbiorów jest takim najliczniejszym zbiorem. Zatem wybierając zbiór o większej mocy, algorytm znajduje najliczniejszy zbiór niezależny T. □

Złożoność

Złożoność czasowa przedstawionego algorytmu to $O(|V|)$, ponieważ wykonuje on po prostu przeszukiwanie w głąb drzewa.