Anna Pacanowska, 306412

Użyjemy lekko zmodyfikowanego DFSa. W każdym wierzchołku v, korzystając z uprzednio policzonych wartości potomków, obliczymy maksymalny zbiór niezależny w poddrzewie z korzeniem v.

```
DFS(v):
    visited[v]=1
    for y ∈ ADJ[x]:
        if visited[y]
            continue
        DFS(y)
        son_set[v] += max_set[y]
        for z ∈ ADJ[y]:
            if v != z:
                grand_set += max_set[z]
    if |son_set| < |grand_set| + 1:
        max_set[v] = grand_set + {v}
    else
        max_set[v] = son_set
```

Poprawność algorytmu udowodnimy przez indukcję po wysokości poddrzewa. Niech max_set[v] oznacza maksymalny zbiór niezależny poddrzewa z korzeniem v.
Oczywiście w liściu l max_set[l] = {l}.
Weźmy dowolny wierzchołek v. Zakładamy, że maksymalne zbiory niezależne wszystkich potomków są już poprawnie obliczone. Mamy dwie możliwości:
1.v $\in$ max_set[v]
Wtedy żaden syn v nie może być w max_set[v], gdyż jest połączony z v krawędzią. Wszystkie wierzchołki w max_set[v] poza v są więc w poddrzewach wnuków. Suma dowolnych zbiorów niezależnych z poddrzew wnuków i v jest zbiorem niezależnym - nie istnieje krawędź między nimi a v, a także te poddrzewa nie są połączone. Warto brać maksymalne zbiory z każdego z tych poddrzew, gdyż te wybory są niezależne. Z założenia indukcyjnego każdy z nich był optymalny. W tym przypadku max_set[v] to suma v i maksymalnych zbiorów z poddrzew wnuków.
2.v $\notin$ max_set[v]
Podobnie jak wcześniej, suma dowolnych zbiorów niezależnych z poddrzew synów jest zbiorem niezależnym. Zatem max_set[v] to suma maksymalnych zbiorów niezależnych z poddrzew synów.
Ostatecznie max_set[v] musi być maksimum z tych dwóch przypadków.

Każda krawędź zostanie odwiedzona dwa razy - raz podczas przechodzenia DFSa, raz podczas wewnętrznej pętli. Algorytm ma więc złożoność $O(n)$.