

Karol Cidyło

Zadanie 6 z listy 2.

Lemat cycle property.

Weźmy cykl  $C$  w grafie. Jeśli waga krawędzi  $e$  z  $C$  jest największą wagą spośród krawędzi cyklu (z pozostałych wierzchołków zbioru  $C$ ) wtedy ta krawędź nie należy do MST.

Dowód:

Załóżmy, że  $e$  należy do MST  $T_1$ , usunięcie  $e$  powoduje cięcie  $T_1$  na dwa poddrzewa z dwoma końcami  $e$  w różnych poddrzewach.

Skoro mieliśmy cykl to możemy znaleźć krawędź  $f$  z  $C$ , która połączy nam nasze początkowe drzewo. Jeśli  $e$  jest maksymalne to  $w(f) < w(e)$ , więc zastępując krawędź  $e$  przez krawędź  $f$  dostajemy MST  $T_2$  o mniejszej wadze niż MST  $T_1$ .

Idea algorytmu: Weźmy krawędź  $e(v_1, v_2)$ . Zapiszmy wagę i usuńmy z grafu. Używając zmienionego DFS przejdziemy krawędzie pod warunkiem, że mają one mniejszą wagę od zapamiętanej wagi krawędzi  $e(v_1, v_2)$ . Jeśli zaczniemy z  $v_1$  i dotrzemy do  $v_2$  to znaczy, że  $e$  należało do cyklu i  $w(e)$  było największą wagą z cyklu (przechodziliśmy tylko krawędzie mniejsze od  $w(e)$ , dodając  $e$  stworzymy cykl). Korzystając z cycle property wiemy, że  $e$  nie należy do MST. Jeśli przejdziemy cały graf (poza krawędzią  $e$ ) i i nie dotrzemy do  $v_2$  to znaczy, że  $e$  należy do MST.

Pseudokod:

```
maks_waga = waga(e)
```

```
dfs(v_start, v_koniec):  
    dla kazdego sasiada u wierzchołka v_start:  
        jesli (u == v_koniec) zwroc falsz  
        jesli odwiedzony[u] == 0 i waga(e(v,u)) < maks_waga  
            dfs(u, v_koniec)  
    zwroc prawda
```

Algorytm działa w czasie  $O(n+m)$  tak jak tradycyjny dfs. Przechodzimy każdą krawędź oraz wierzchołek raz.

Może nawet zakończyć się szybciej, jeśli dojdziemy do wierzchołka  $v_2$ .

Poniżej zamieszczam dokładniejszą implementację wykorzystującą listę sąsiedztwa wierzchołków w grafie oraz iteracyjny dfs-a, według mnie jest to najlepsze rozwiązanie, ponieważ pozwala łatwo zakończyć działanie po dotarciu dożądanego wierzchołka.

Przepraszam za niespójność w języku wyrażen. Niżej używam angielskiego, ponieważ kod naturalniej pisze mi się używając angielskich nazw oraz zwrotów.

```

dfs(e):
    max_weight = weight(e)
    stack.push(e.start)

    while (!stack.empty())
    {
        s = stack.top();
        stack.pop();

        if (!visited[s])
        {
            visited[s] = true;

            for (auto i = adj[s].begin(); i != adj[s].end(); ++i)
                if (*i == e.end) return false;
                if (!visited[*i] && weight(s, *i) < max_weight)
                    stack.push(*i);
            }
        }

    return true;
}

```