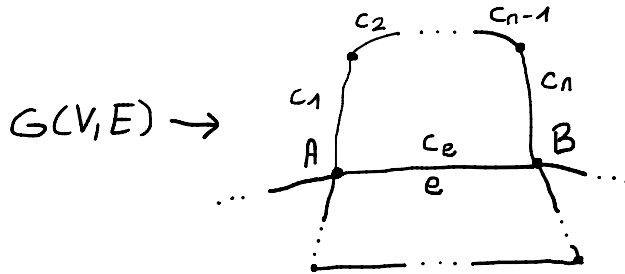


Zad 6.

poniedziałek, 20 marca 2023 22:42

6. (2pkt) Ułóż algorytm, który dla danego spójnego grafu G oraz krawędzi e sprawdza w czasie $O(n + m)$, czy krawędź e należy do jakiegoś minimalnego drzewa spinającego grafu G . Możesz założyć, że wszystkie wagi krawędzi są różne.



- 1) jeżeli e jest mostem, to na pewno należy do każdego MST.
 - 2) jeżeli nie, to $\forall \text{MST } e \notin \text{MST}$
- \Updownarrow
 $e \in C\text{-cykl i}$
 $\forall f \in C \setminus e \exists c(f) < c(e)$

algorytm:

```

procedure dfs(u):
    visited[u] = true
    dla każdego sąsiada v wierzchołka u:
        jeżeli !visited[v] oraz  $c(u, v) < c(e)$ :
            dfs(v)
    
```

```

procedure is_in_mst( $e = \{u, v\}$ ):
    dfs(u)
    jeżeli visited[v] = true:
        zwroc "NIE"
    wpp. zwroc "TAK"
    
```

dowód poprawności:

"circle property"

jeżeli dfs puszczony z wierzchołka u nie odwiedzi wierzchołka v , to można wydalizyć się dwie rzeczy:

tu nie wystarczy, ~~możemy~~ mogłyby wydać się dwie rzeczy:

1) e jest mostem i wtedy e należy do każdego MST

2) e nie jest mostem $\Rightarrow e$ należy do jakiegoś cyklu (cykli) i na żadnym z nich nie jest krawędzią o maksymalnej wadze. Wtedy każde drzewo ~~spinające~~ T , po dodaniu do niego e posiada jeden z tych cykli. Usuwając krawędź o większej wadze dostajemy znowu drzewo ~~spinające~~ o wadze mniejszej niż T , zatem $\forall \text{MST } e \in \text{MST}$

nie wprost
 $e \notin T$

"minimalne"

□