MIN KOSTRY #2

$G=(V,E,w)$ $\quad w: E \to \mathbb{R}$, BÚNO G souvislý

Kruskalův (hladový / greedy) algoritmus $\quad \boxed{\Theta(m \log n)}$

-1/2) INIT-UF(V) $\quad$ Pamatuj $\quad \Theta(m)$ $\quad$ zřízení UNION-FIND $\quad \Theta(n)$

0) $T := \emptyset$

1) SORT E dle w $\quad \Theta(m \cdot \log m) = \Theta(m \cdot \log n)$

2) for $uv \in E$ (v pořadí dle w)

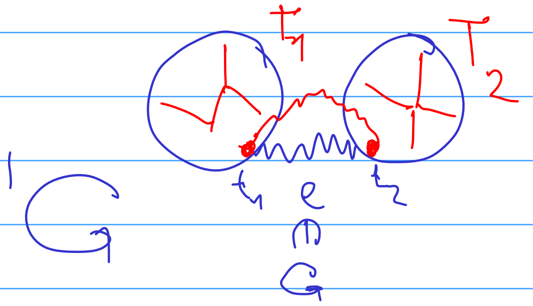
- if NOT FIND(u,v): $\quad O(m \cdot \log n)$
  - $T := T + uv$
  - UNION(u,v) $\quad O(n \cdot \log n)$

UNION-FIND $\Theta(\log n)$ stromová struktura

Kruskalův alg vrátí kostru.

1) z def T je acyklický

2) kdyby T nebyl souvislý graf na V tak že souvislosti G



T $\quad \exists e \in E$ t.ž. $e = t_1 t_2$, $t_1 \in T_1$, $t_2 \in T_2$

Ale Kruskal nepřidal e do $T \Rightarrow$

T musí obsahovat cestu z t_1 do t_2 ↯

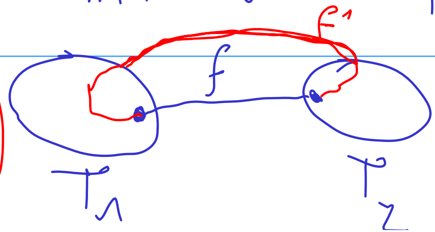
Věta: T (z Kruskalova alg) je MST

Lemma: $G=(V,E,w)$ je graf, C je cyklus v G, a $f \in E(C)$ t.ž. $w(f) = \max_{e \in E(C)} w(e)$ tak pak existuje T min. kostra že $f \notin T$. Pokud $\exists$ MST T_0 t.ž. $f \in T_0$, tak $\underbrace{\exists f' \in E(C)\setminus\{f\}}$, $T = T_0 - f + f'$ je MST

Dk (lemmatu): vezmi nějakou MST T_0. Pokud $f \notin T_0$ OK.

$T_0 - f$ má dvě komponenty T_1 a T_2

$\boxed{T := T_0 - f + f'}$



$\exists f' \in E(C)$ t.ž. vede mezi T_1 a T_2 $\Rightarrow T_0 - f + f'$ je strom. $w(f) \geq w(f')$

Dk (Kruskal): $T :=$ kostra z ALG

nechť pro spor $\exists T_0$: $\sum_{e \in T_0} w(e) < \sum_{e \in T} w(e)$

ze všech takových T_0, zvol takové, že $|E(T) \cap E(T_0)|$ je max možné

$f :=$ hrana $E(T_0) \setminus E(T)$ s nejmenší možnou vahou

$w(f) = \min_{e \in E(T_0) \setminus E(T)} w(e)$

V ALG v kroku s hranou f existoval C v $T + f$

$\forall e \in E(C)$, $w(e) \leq w(f)$

Tak pak ale z lemmatu pro $T_0, f, C \rightsquigarrow$ ex. MST T_0' že $T_0' = T_0 - f + f'$ $f' \in E(C) \setminus \{f\}$

Ale T_0' je MST, neobsahuje $f \notin T$ a obsahuje $f' \in T$

$\lightning$ s max $|E(T) \cap E(T_0)|$

$\boxed{A}$

UNION-FIND struktura na množině V

CÍL: držet rozklad V na části $V_1 \dots V_k$

OPERACE: Find(u,v) < ANO $\exists i$ t.ž. $\{u,v\} \subseteq V_i$; NE jinak

Union(u,v): spoj část obsahující u s částí obsahující v

---

pole UF indexovat prvky V

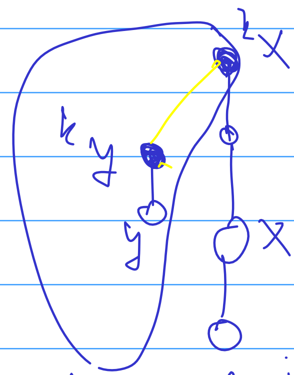
$UF[u] \geq 0$: $UF[u]$ je index předchůdce u v UF

$UF[u] < 0$: u kořen v UF a $-UF[u]$ je hloubka keříku s kořenem u

---

```
kořen(x):
    y := x
    while UF[y] ≥ 0:
        y := UF[y]
    return y
```

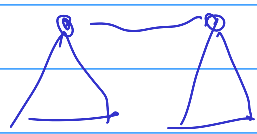
$\Theta(\text{hloubka}) = O(\log n)$



$h_y + 1 \leq h_x$

FIND(x,y): return $\boxed{\text{kořen}(x)} = \boxed{\text{kořen}(y)}$

```
UNION(x,y):
    k_x := kořen(x), k_y := kořen(y)
    if k_x = k_y: RETURN
    else:
        h_x := -UF[k_x], h_y := -UF[k_y]
        if h_x > h_y: UF[k_y] = k_x
        else if: h_y > h_x: UF[k_x] = k_y
        else: UF[k_y] = k_x
              UF[k_x] = UF[k_x] - 1
```

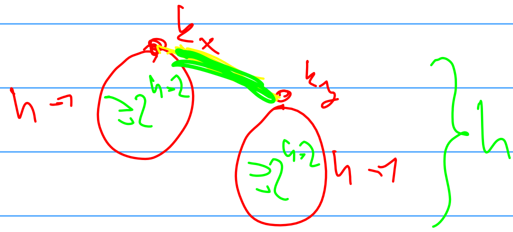


INIT(V): $\forall u$ $UF[u] = -1$

LEMMA: keřík hloubky h obsahuje $\geq 2^{h-1}$ prvků

Dk: indukcí dle h. $h=1$ ✓

$h>1$, podíváme se na situaci kdy h zvětšilo o 1



$\rightsquigarrow \geq 2^{h-1}$ protože $\boxed{}$