Iskanje najdaljse poti na kaktusih

Vid Treven

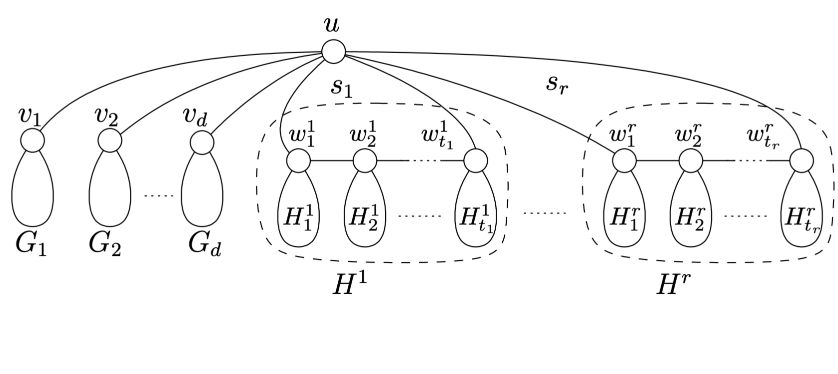
December 2022

1 Potrebne definicije

Kaktus (tudi kaktusovo drevo) je povezan graf, za katerega velja da imata katerakoli dva cikla tega grafa najvec eno skupno vozlisce. Ekvivalentna definicija se glasi: Kaktus je povezan graf, katerega vsaka povezava je v najvec enemu ciklu. **Korenski kaktus** je kaktus, ki ima eno izmed vozlisc za koreno.
Ciklicna vrsta je vrsta A[1,...,n], za katero velja:

1. $n \geq 3$
2. $A[1]$ in $A[n]$ sta soseda v tej vrsti

V kontekstu ciklicnih vrst je **razdalja** definirana kot
D$(i, j) = \max \{|i - j|, n - |i - j|\}$
Naj bo G korenski kaktus s korenom u. Recimo, da je koreno u vozlisce v r ciklih $s_1, s_2, ..., s_r$. Ce izbrisemo koreno u dobimo $d + r$ povezanih komponent. Te komponente so **minion-i**. d minion-om, ki so z vozliscem u povezani z enojnimi povezavami pravimo **drevesni minion-i**, ostalim r pa **ciklicni minion-i**. Vsak drevesni minion je spet korenski kaktus, cigar koreno je vozlisce, povezano z u-jem. To pa ne velja za ciklicne minione. V vsakem ciklicnem minionu H_i je pot $s_i - u$ **baza** (V nasem primeru je baza za H_1 je $w_1^1, w_2^1, ..., w_{t_1}^1$). Korena drevesnih minionov so **drevesni otroci u-ja**. Drevesni otroci ter vozlisca v bazah ciklicnih minionov so **otroci u-ja** (V nasem primeru so drevesni otroci u-ja $v_1, ..., v_d$, otroci u-ja pa so $v_1, ..., v_d, w_1^1, ..., w_{t_1}^1, ..., w_1^2, ..., w_{t_2}^2$). Vozliscu brez otrok pravimo **list**.



2 Algoritem

Algoritem za iskanje najdaljse poti v kaktusih deluje po principu deli in vladaj, torej velik problem se razdeli na vec manjsih problemov, ki so lazje resljivi. Osnovni algoritem NAJDALJSI KAKTUS izbere koren u in poklice funkcijo NAJDALJSI KORENSKI KAKTUS. Ta algoritem vrne dve stevili. Prvo ($l_1(u)$) predstavlja dolzino najdaljse poti v podkaktusu s korenom u, drugo ($l_2(u)$) pa je enako najdaljsi poti v podkaktusu s korenom u, katere koncno vozlisce je u. Algoritem deluje tako, da zacetni graf razdeli na manjse podgrafe (minione) in na njih spet izvede algoritem NAJDALJSI KORENSKI KAKTUS. Za ciklicne minione uporabimo se algoritem AUX, ki v ciklicni vrsti poisce najdaljso pot (razdaljo med dvema vozliscema + njuno l_2).

Algorithm 1 NAJDALJSI KAKTUS(G)

```
Izberi katerokoli vozlisce u na grafu G
Predstavi graf G kot korenski kaktus s korenom u
NAJDALJSI KORENSKI KAKTUS(G[u], u)
return l_1(u)
```

Algorithm 2 NAJDALJSI KORENSKI KAKTUS(G[u], u)

if u je list **then**
    $(l_1(u), l_2(u)) = (0, 0)$
else
    drevesni otroci u-ja so $v_1, ..., v_d$
    ciklicni minioni u-ja so $H^1, ..., H^r$
    baza za H^i je $w_1^i, w_2^i, ..., w_{t_1}^i$
    for u-jev otrok a **do**
        NAJDALJSI KORENSKI KAKTUS(G[a], a)
    end for
    $P = \{l_2(v_i) + 1 | 1 \le i \le d\}_M$
    for $1 \le i \le r$ **do**
        $q_i = max\{l_2(w_k^i) + max\{k, t_i - k + 1\} | 1 \le k \le t_i$
    end for
    $Q = \{q_i | 1 \le i \le r\}_M$
    for $1 \le i \le r$ **do**
        $z_i = \text{AUX}(A[0, l_2(w_1^i), ..., l_2(w_{t_1}^i)])$
    end for
    $x = \max\{P \cup Q\}_M$
    $y = \text{second-max}\{P \cup Q\}_M$
    $z = \max\{z_i | 1 \le i \le r\}_M$
    $m = \max\{l_1(a) | a \text{ je otrok } u\text{-ja}\}_M$
    $l_1(u) = \max\{x + y, z, m\}$
    $l_2(u) = x$
end if

Algorithm 3 AUX(A[1...n])

$B[0...n]$ in $C[1...n]$ naj bosta linearni vrsti
$B[0] = 0$
for $1 \le i \le n$ **do**
    $B[i] = max\{B[i-1], A[i] - (i-1)\}$
    $C[i] = B[i-1] + A[i] + (i-1)$
end for
$x = \max\{C[i] | 1 \le i \le n\}$
for $1 \le i \le n$ **do**
    $B[i] = max\{B[i-1], A[i] + (i-1)\}$
end for
$C[n] = A[n] + 1$
for $1 \le i \le n$ **do**
    $C[i] = max\{C[i+1], A[i] + n - (i-1)\}$
end for
$y = \max\{B[i] + C[i+1] | 1 \le i \le n-1\}$
return $\max\{x, y\}$