アルゴリズム 2

Algorithm Find Innermost s-sided Cut of Minimum Size

初期設定:

$g(s), g(t) \in \mathbb{N}$

点素 s-t パス $f_1, ..., f_k$

$\ell_i : f_i$ の長さ ((s) は 0 番目)

$v_{ij}, 0 \le j \le \ell_i$ を f_i 上の j 番目の頂点とする

$w_i = v_{i1}(1 \le i \le k)$

valid := **false**

for $i = 1$ to k **do**
    $c := 0$ ▷ f_i の始まり
    $d := \ell_i$ ▷ f_i の終わり
    while $d \neq c + 1$ **do**
        $e := \lceil \frac{c+d}{2} \rceil$ ▷ f_i 二分探索
        $W := w_1, ..., w_{i-1}, v_{ie}, w_{i+1}, ..., w_k$ ▷ i 番目を動かす
        if $W^+ \neq (t, t, ..., t)$and$|V_s(W^+) + g(s) \le |V_r(W^+) \cup V_t(W^+)| + g(t)$ **then**
            $c := e$
            valid := **true**
        else
            $d := e$
        end if
    end while ▷ $O(\log n)$ 回反復
    $w_i := v_{ic}$
end for
if vaild **then**
    return $(w_1, ..., w_k)$
else
    return $(s, s, ..., s)$
end if

アルゴリズム 3

Algorithm Find Innermost t-sided Cut of Minimum Size

初期設定:

$g(s), g(t) \in \mathbb{N}$

点素 s-t パス $f_1, ..., f_k$

ℓ_i : f_i の長さ ((s) は 0 番目)

$v_{ij}, 0 \leq j \leq \ell_i$ を f_i 上の j 番目の頂点とする

$w_i = v_{i1} (1 \leq i \leq k)$

valid := **false**

for $i = 1$ to k **do**
  $c := 0$ ▷ f_i の始まり
  $d := \ell_i$ ▷ f_i の終わり
  while $d \neq c + 1$ **do**
    $e := \lfloor \frac{c+d}{2} \rfloor$ ▷ f_i 二分探索
    $W := w_1, ..., w_{i-1}, v_{ie}, w_{i+1}, ..., w_k$ ▷ i 番目を動かす
    if $W^- \neq (t, t, ..., t)$and$|V_t(W^-) + g(t) \leq |V_r(W^-) \cup V_s(W^-)| + g(s)$ **then**
      $d := e$
      vaild := **true**
    else
      $c := e$
    end if
  end while ▷ $O(\log n)$ 回反復
  $w_i := v_{id}$
end for
if vaild **then**
  return $(w_1, ..., w_k)$
else
  return $(s, s, ..., s)$
end if

アルゴリズム 2,3 共に $O(km \log n)$ 時間

分散案

補題 1 (経路収集)**.** G の有向パス $P = \{v_1, ..., v_\ell\}$ を考える。ここで、各ノード v_i はそのパスの先頭ノードと後尾ノードを知っている。$O(\log n)$SA ラウンドにおいて、各ノード v_i は i の値と経路内のそのインデックスを知ることができる。さらに、各ノード v_i が整数 x_i と共通の結合演算子 $\oplus$ を知っていれば、各ノード v_i に先頭集約 $\bigoplus_{j \leq i} xj$ と後尾集約 $\bigoplus_{j \geq i} x_j$ を学習させることができる。

パスの長さやインデックスは $O(\log n)$SA ラウンドで計算可能. 二分探索はパス上のノードだけが計算すればよさそう？