# 6.图

## Prim算法

## 极短跨边

邓俊辉

deng@tsinghua.edu.cn

# 割 & 极短跨边

❖ 设(U; V\U)是N的割 cut

❖ 【Cut Property-A】

若：(u, v)是该割的极短跨边（shortest crossing edge）

则：必存在一棵包含(u, v)的MST

❖ 反证：假设(u, v)未被任何MST采用...

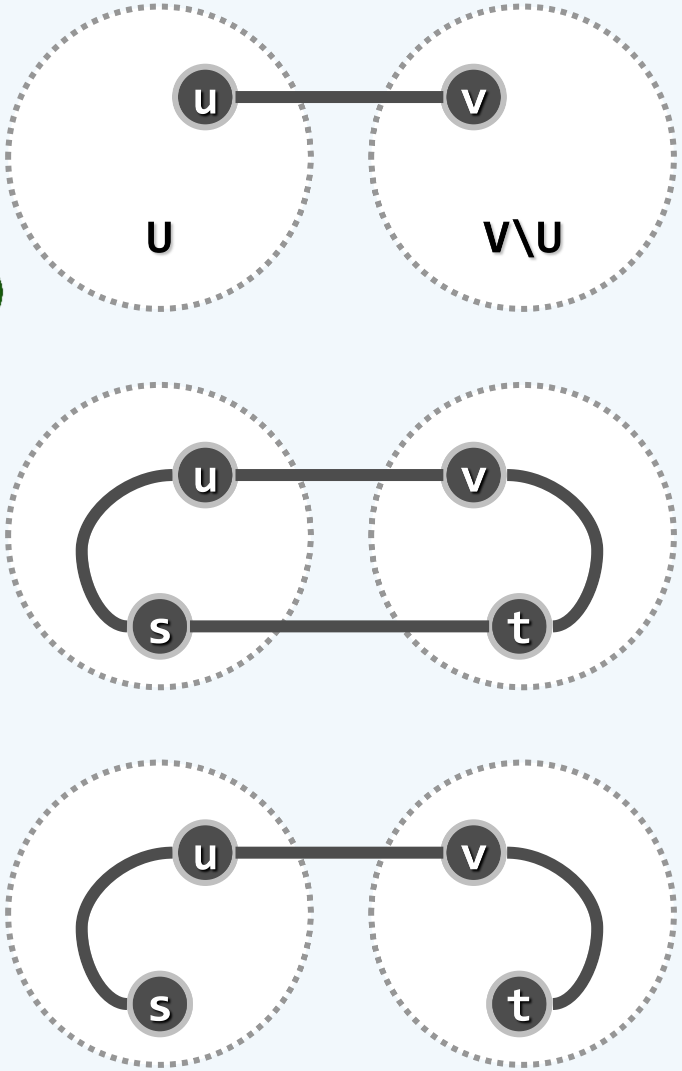
❖ 任取一棵MST，将(u, v)加入其中，于是

将出现唯一的回路，且该回路必经过

(u, v)以及至少另一跨边(s, t)

❖ 现在，将原MST中的(s, t)替换为(u, v)...

❖ 【Cut Property-B】反之，N的任一MST也必通过极短跨边联接每一割



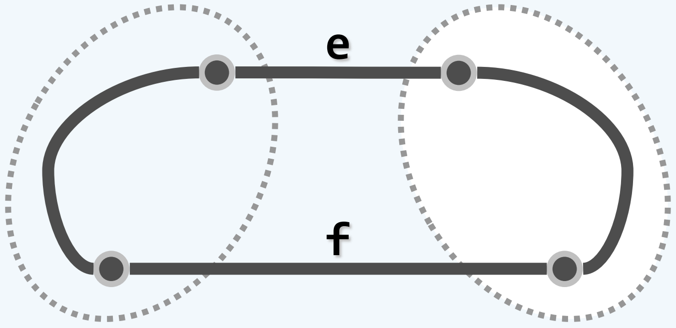
# 环 & 极长环边

❖ 设T是N的一棵MST，且在N中添加边e后得到N'

❖ 【Cycle Property】

若：沿着e在T中对应的环路，f为一 极长边

则：T - {f} + {e}即为N'的一棵MST




❖ 1)若e为环路上的 最长边 ，则与前同理，e不可能属于N'的MST

此时，f = e，T - {f} + {e} = T依然是N'的MST

❖ 2)否则有：|e| ≤ |f|；移除f后T - {f}一分为二，对应于N/N'的割

在N/N'中，f/e应是该割的 极短跨边

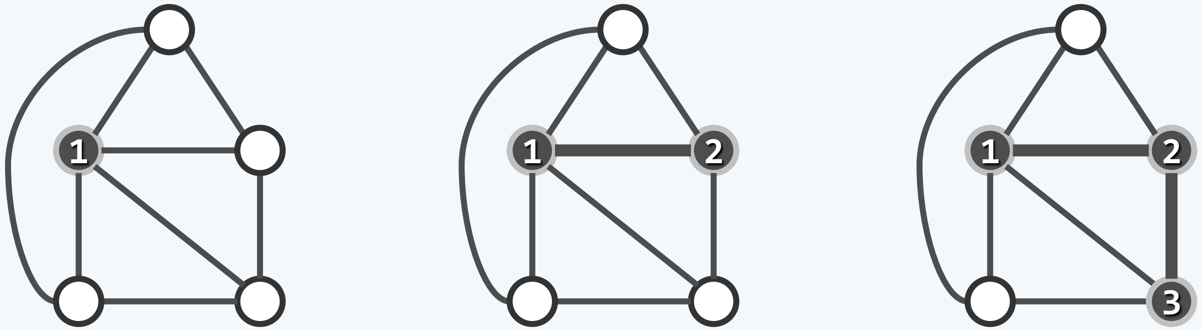
此割在N和N'中导出的一对互补子图完全一致

故，这对子图各自的MST经e联接后，即是N'的一棵MST

# 算法

❖ 从$\boxed{T_1}$ = ( {$v_1$}; ∅ )开始，逐步构造$\boxed{T_2}$、$\boxed{T_3}$、...、$\boxed{T_n}$，其中

- $v_1$可以任选

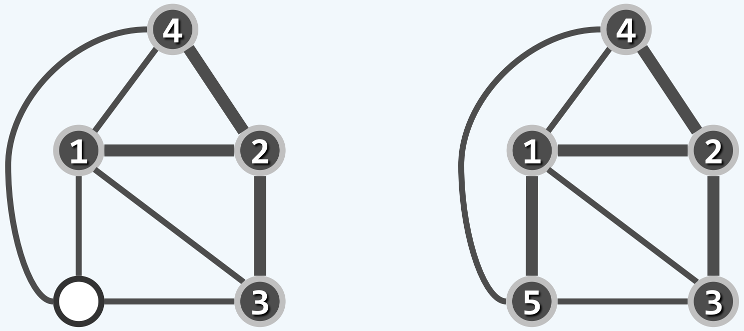



- $T_k = ( V_k; E_k )$

  $|V_k| = k$ , $|E_k| = k-1$

  $V_k \subset V_{k+1}$

❖ 由以上分析，为由$\boxed{T_k}$构造$\boxed{T_{k+1}}$，只需




- 将($V_k$ : $V \backslash V_k$)视作原图的一个割

- 在该割的所有跨边中，找出$\boxed{\text{极短者}}$ $e_k = ( v_k, u_k )$

- 令$T_{k+1} = ( V_{k+1}; E_{k+1} ) = ( V_k \cup \{u_k\}; E_k \cup \{e_k\} )$