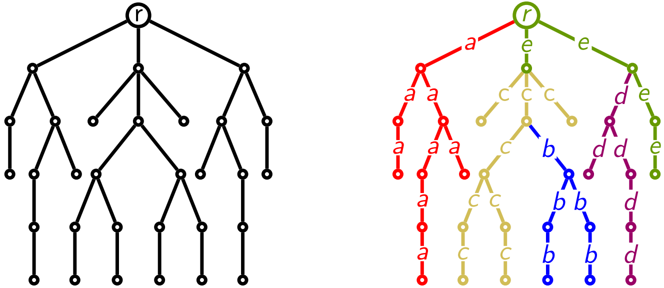
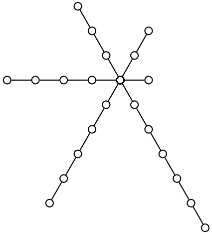
k-split



$$\text{EMax}(\mathcal{P}) = \_,\ \text{EMin}(\mathcal{P}) = \_,\ \text{ratio}(\mathcal{P}) = \_,\ \text{diff}(\mathcal{P}) = \_.$$

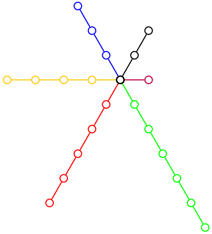
Vertex Partition vs edge Partition

R. Becker *et. al*(JACM, 1982,1983) gave a poly-time algorithm for min-max and max-min objects for vertex partition problem on a tree.



G. N. Frederickson *et. al*(SODA, 1991) gave a linear-time algorithm for min-max and max-min objects for vertex partition problem on a tree.

B.Y. Wu *et. al*(DAM, 2007) showed that the min-max, max-min and min-ratio objects for edge-partition problem on a tree is NP-hard.



Related Result

$\mathcal{P}$ is a k-split of a tree. B.Y. Wu showed that:
min(ratio($\mathcal{P}$)) $\leq$ _, $\forall k$ in ________
min(ratio($\mathcal{P}$)) $\leq$ _, for $k =$ _____

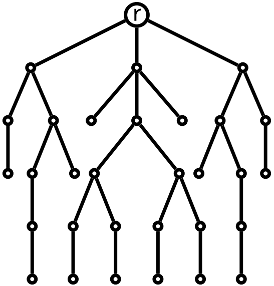
Conjecture (Wu's k-split conjecture)

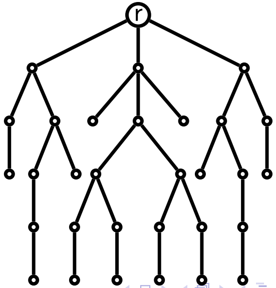
Given a tree T with n edges and a positive integer $k \leq n$, we can always split T into k edge-disjoint subtrees such that the ratio of the maximum number to the minimum number of edges of the subtrees is at most two.

Split lemma

Lemma (splitting lemma)

*Let P be a tree with root r. For any $2 \le \lambda \le e(P)$, we can find a subtree P' in linear time, such that $e(P')$ is in a closed interval $[\lambda, 2\lambda - 2]$. Moreover, when we remove P' from P, we get a subtree of P containing the root r.*Remark: λ 的上限仍可維持 $e(P)$ ，當 $\lambda > \frac{e(P)}{2}$ 時，則 P' 取整個 P ，但 remove P' 後，P 仍剩一點 root。



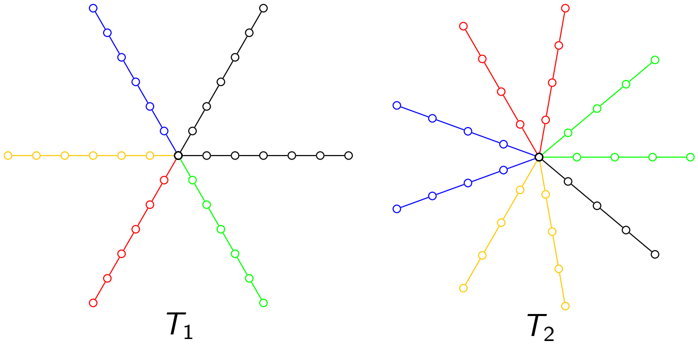


Where is the boundary?

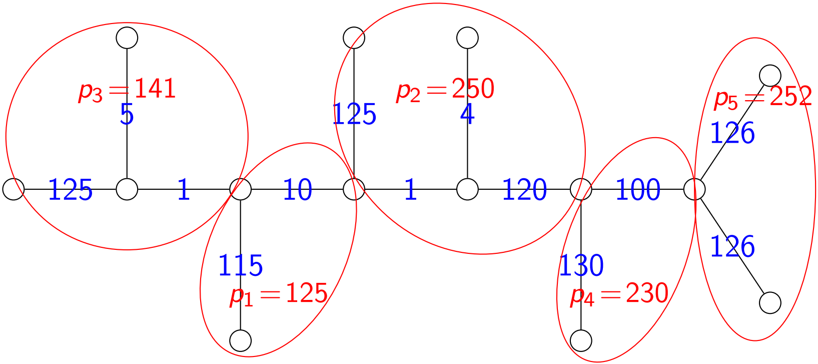
EMax($\mathcal{P}$) $\leq$ ____, ex: $e(\mathcal{P}) = (1, 1, \cdots, 1, 2)$

EMin($\mathcal{P}$) $\geq$ ____, ex: $e(\mathcal{P}) = (1, 2, 2, \cdots, 2)$

When n=36, $\exists$ T_1 s.t. min(EMax($\mathcal{P}$)) = 12.
When n=36, $\exists$ T_2 s.t. max(EMin($\mathcal{P}$)) = 4,

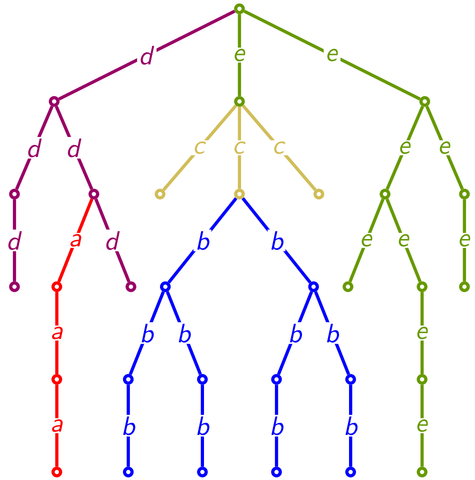


Split a part with weight closest to the average



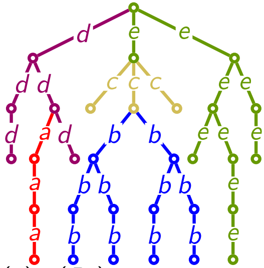
i	5	4	3	2	1
n_i	998	746	516	375	125
m_i	199.6	186.5	172	187.5	125
p_i	252	230	141	250	125

When ratio is greater than 2, . . .

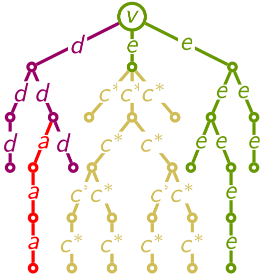


Key:_______

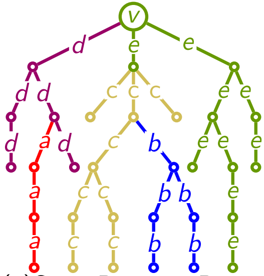
$e(P_b) = 10, e(P_a) = e(P_c) = 3.$



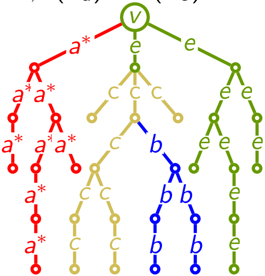
(a) $e(P_b) = 10, e(P_a) = e(P_c) = 3$.



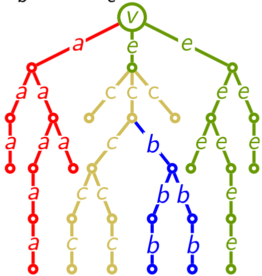
(b) Combine P_c with P_b into P_{c^*}.



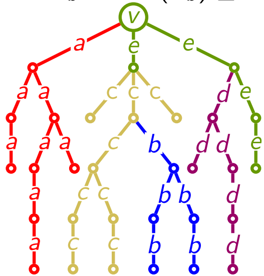
(c)Split P_{c^*} into P_c and P_b with $e(P_b) \geq 5$.



(d)Combine P_a and P_d into P_{a^*}.



(e) $e(P_{a^*}) < e(P_e)$, $P_a \leftarrow P_{a^*}$.



(f) Split $e(P_e)$ into P_d and P_e.

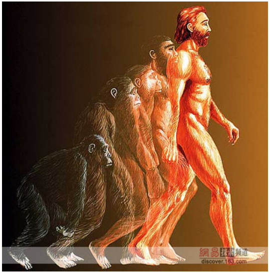
Evaluate-And-Evolve

For a k-split $\mathcal{P}$ of a tree, if ratio($\mathcal{P}$) > 2, we execute $\textsc{Evolution}(\mathcal{P})$ to get a better $\mathcal{P}$.

The idea of $\textsc{Evolution}$ is to “balance” the size of different parts. Move some edges from the largest part to the smallest one.

To Claim the processes would terminal in finite steps: Use an integer function Eval : $\mathcal{P} \to \mathbb{N}$ and claim

$$\text{Eval}(\textsc{Evolution}(\mathcal{P})) < \text{Eval}(\mathcal{P}).$$



Evaluate Partition

Definition

$DMax(\mathcal{P}) = EMax(\mathcal{P}) - \lceil \frac{n}{k} \rceil$, $DMin(\mathcal{P}) = \lfloor \frac{n}{k} \rfloor - EMin(\mathcal{P})$.
$NMax(\mathcal{P}) = |\{P_i : e(P_i) = EMax(\mathcal{P})\}|$,
$NMin(\mathcal{P}) = |\{P_i : e(P_i) = EMin(\mathcal{P})\}|$.

$$EEval(\mathcal{P}) = (DMax(\mathcal{P}), NMax(\mathcal{P}), DMin(\mathcal{P}), NMin(\mathcal{P})).$$

The order of $EEval(\{P_i\})$ is compared by lexicographic order.

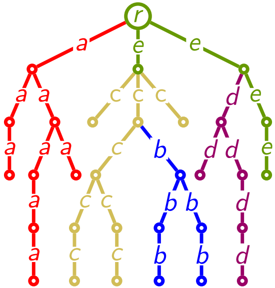
We have ________ $<$ EEval($\mathcal{P}$) $<$ __________, for any $\mathcal{P}$.
For example, $n = 100, k = 10$,
$e(\mathcal{P}_1) = (5, 5, 5, 6, 9, 12, 14, 14, 15, 15)$,
$e(\mathcal{P}_2) = (4, 5, 5, 9, 9, 12, 14, 14, 14, 14)$.
We have EEval($\mathcal{P}_1$) $= (5, 2, 5, 3) >$ ________ $=$ EEval($\mathcal{P}_2$).

Partition Path

Definition

$\mathcal{P} = \{P_1, P_2, \cdots, P_k\}$ *is a k-split of a tree T. For any two parts* $P_i, P_j \in \mathcal{P}$, *we define* $PPath(P_i, P_j) = \{P_{p_1}, P_{p_2}, \cdots, P_{p_m}\}$ *as follows:*

1. $p_1 = i$, $p_m = j$
2. *For any two vertices* $v_i \in P_i, v_j \in P_j$, $\{P_{p_2}, \cdots, P_{p_{m-1}}\}$ *are the parts which we cross in order when traversing along the path from* v_i *to* v_j.



Balance Lemma

```
BALANCE({P_{p_1}, ..., P_{p_m}}, λ)
i = 1
while P_{p_i} < λ and i < m
      do P*_{p_i} = P_{p_i} ∪ T_{p_{i+1}}
         P_{p_i} = SPLIT(P*_{p_i}, λ)
         P_{p_{i+1}} = P*_{p_i} \ P_{p_i}
         i = i + 1
```

Lemma (Balance Lemma)

$\mathcal{P}$ is a k-split of tree T. Assume that $ratio(\mathcal{P}) > 2$, $e(P_{\min}) = EMin(\mathcal{P})$ and $e(P_{\max}) = EMax(\mathcal{P})$. Let $\mathcal{P}'$ be the partition after executing $\text{BALANCE}(PPath(P_{\min}, P_{\max}), \underline{EMin(\mathcal{P}) + 1})$. We have $EEval(\mathcal{P}') < EEval(\mathcal{P})$.

Proof of Balance lemma :

Proof:

BALANCE executes i times while-loop.

- $i < m$:
  DMax($\mathcal{P}'$), NMax($\mathcal{P}'$) don't increase.
  $\text{EMin}(\mathcal{P}) + 1 \leq e(P'_{p_t}) \leq 2 \cdot \text{EMin}(\mathcal{P})$, $\forall 1 \leq t < i$,
  and $P_{p_1} < \text{EMin}(\mathcal{P}) + 1 \leq P'_{p_1}$.
  Hence, EEval($\mathcal{P}'$) decreases.
- $i = m$:
  $\text{EMin}(\mathcal{P}) + 1 \leq e(P'_{p_t}) \leq 2 \cdot \text{EMin}(\mathcal{P})$, $\forall 1 \leq t < m$,
  $T_{p'_m} < P_{p_m}$.
  Hence, EEval($\mathcal{P}'$) decreases.

□

k-split in $O(n^3)$ time.

Theorem

For any tree T with $e(T) = n$, there is an algorithms to get a k-split $\mathcal{P}$ with ratio$(\mathcal{P}) \leq 2$ in $O(n^3)$.

Proof:

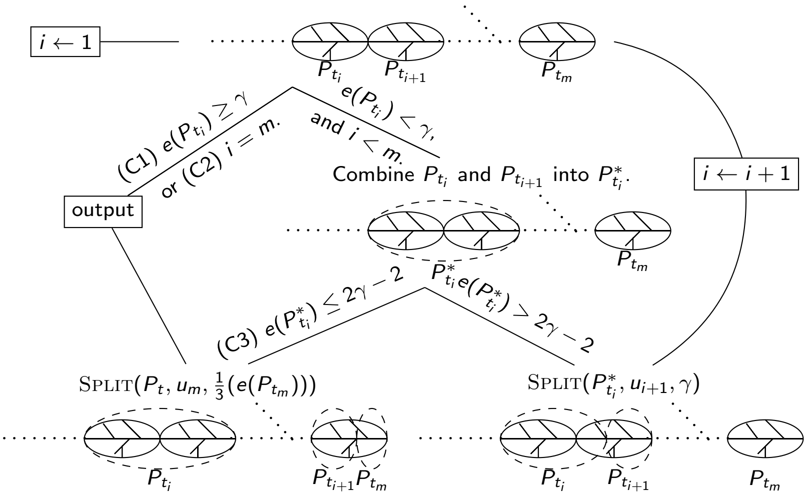
Choosin a $\mathcal{P}$ with EMax$(\mathcal{P}) \leq \frac{2n}{k+1} < 2\frac{n}{k}$.

$\sharp$ (values smaller than EEval$(\mathcal{P})$) $\leq$ ____________ .

By Balance Lemma, after executing BALANCE at most $2n^2$ times, we have ratio$(\mathcal{P}) \leq 2$.

In BALANCE, we use SPLIT at most k times. And $e(T^*_{p_i}) \leq$ __ . So the time complexity of BALANCE is $O(3\frac{n}{k} \cdot k) = O(n)$. Hence, this algorithm takes in $O(n^3)$ time. □

k-split in $O(n^2)$ time : New Balance.



k-split in $O(n^2)$ time : New Evaluation.

$\mathrm{Eval}(\mathcal{P}) = k \cdot (\mathrm{EMax}(\mathcal{P}) - \lceil \frac{n}{k} \rceil) - \mathrm{NAccept}(\mathcal{P})$,
where $\mathrm{NAccept}(\mathcal{P})$ is the number of parts with size in the range
$[\lfloor \frac{\mathrm{EMax}(\mathcal{P})+1}{2} \rfloor, 2\lfloor \frac{\mathrm{EMax}(\mathcal{P})+1}{2} \rfloor - 2]$.

Lemma (NewBalance Lemma)

$\mathcal{P}$ is a k-split of tree T. Assume that $ratio(\mathcal{P}) > 2$, $e(P_{\min}) = EMin(\mathcal{P})$ and $e(P_{\max}) = EMax(\mathcal{P})$. Let $\mathcal{P}'$ be the partition after executing $\textsc{NewBalance}(PPath(P_{\min}, P_{\max}), \underline{\lfloor \frac{EMax(\mathcal{P})+1}{2} \rfloor})$. *We have $Eval(\mathcal{P}') < Eval(\mathcal{P})$.*

Theorem

For any tree T with $e(T) = n$, there is an algorithms to get a k-split $\mathcal{P}$ with $ratio(\mathcal{P}) \leq 2$ in $O(n^2)$.