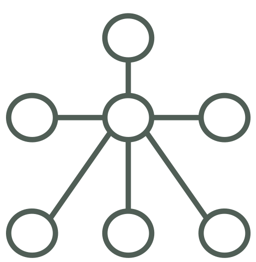
Parameterized Algorithm 7.2, 7.3
tree-width と重み付き最大独立点集合

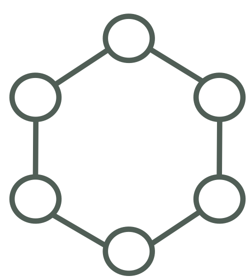
北海道大学 情報知識ネットワーク研究室 M1
大泉翼

木幅とは

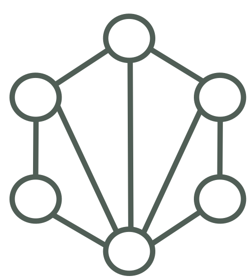
- **木幅（tree-width）：**
  無向グラフに対して定義される不変量の一つ
  - 大雑把にいうと，グラフの**木っぽさ**を表す指標
  - 木幅が小さいほどグラフは**木っぽい**



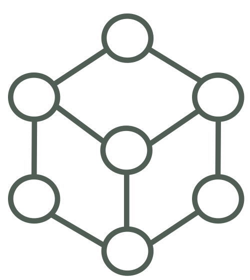
木：$\mathrm{tw} = 1$



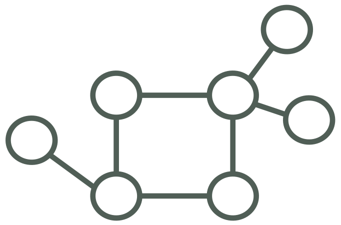
サイクル：$\mathrm{tw} = 2$



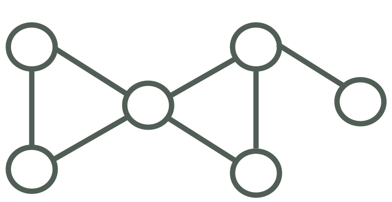
外平面グラフ：$\mathrm{tw} \leq 2$



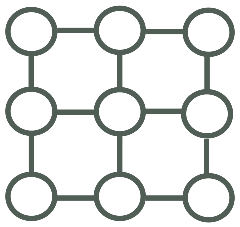
平面グラフ：$\mathrm{tw} = O(\sqrt{n})$



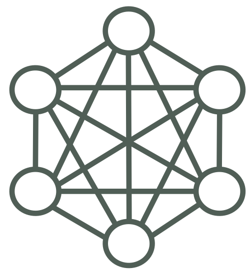
擬似木：$\mathrm{tw} = 2$



カクタス木：$\mathrm{tw} \leq 2$



グリッド：$\mathrm{tw} = \sqrt{n}$



完全グラフ：$\mathrm{tw} = n - 1$

木に対する動的計画法

- 一般グラフでは NP 困難な問題例：
  - 最大独立点集合問題（maximum independent set problem）
  - 最小点被覆問題（minimum vertex cover problem）
  - など

- グラフが木のとき，
  動的計画法を用いることで多項式時間で解けることが多い

- グラフが**木っぽい**とき，つまり木幅が定数で抑えられるとき，
  上記と同様に多項式時間で解くことができる

➡ 本スライドでは，重み付き最大独立点集合を紹介

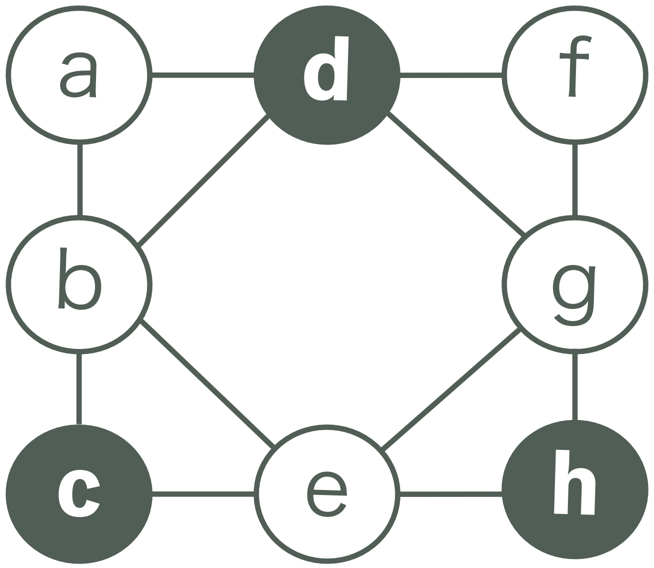
本スライドで扱う問題

重み付き最大独立点集合問題
(Maximum Independent Set Problem)

入力：無向グラフ $G = (V, E)$ と頂点の重み関数 $w : V \to \mathbb{R}$

出力：**独立点集合**のうち，重み和が最大のもの

独立点集合：
V の部分集合のうち，どの 2 頂点も辺で結ばれていないようなもの

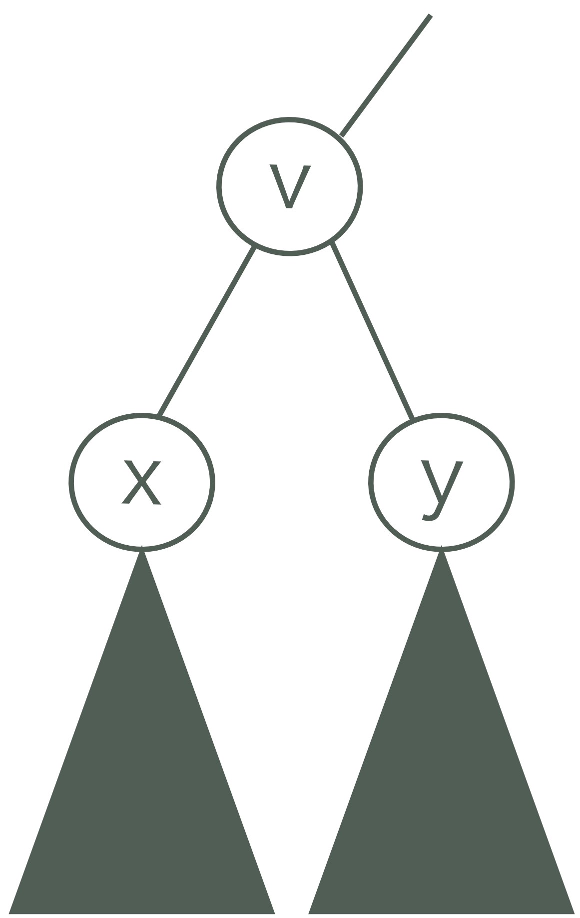


頂点	a	b	c	d	e	f	g	h
重み	3	5	2	8	1	4	3	9

重み和の最大値：19

グラフが木のとき

- 重み付き最大独立点集合問題は，入力グラフが木のとき，
  根付き木をボトムアップに DP することで，$O(n)$ 時間で解ける



T_v： v を根とする寝付き木

DP テーブルの定義：

$A[v] :=$ T_v のに対する解

$B[v] := T_v/\{v\}$ に対する解

定義の気持ち：
頂点 v を含むか
含まないかで
場合分け

DP 遷移式：

$$B[v] = \sum_{c \in child(v)} A[c]$$

$$A[v] = \max \left\{ B[v], w(v) + \sum_{c \in child(v)} B[c] \right\}$$

木分解の定義

$G=(V,E)$ の**木分解（tree-decomposition）**とは,

木 T と**バッグ** $X_t \subseteq V(G)$ からなる集合の組 $\left(T, \{X_t\}_{t \in V(T)}\right)$

であり，以下の性質を満たすものである.

（T1） $\displaystyle\bigcup_{t \in V(T)} X_t = V(G)$

（T2） 全ての辺 $uv \in E(G)$ に対して,

ノード $t \in V(T)$ が存在して $u, v \in X_t$

（T3） 全ての頂点 $v \in V(G)$ に対して,

v を含むバッグに対応するノードの集合は T で連結

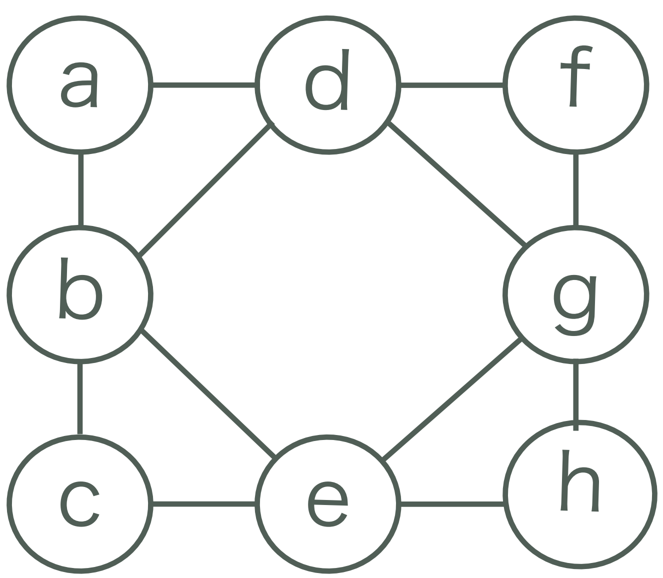
木分解の例（1/2）

（T1） $\bigcup_{t \in V(T)} X_t = V(G)$

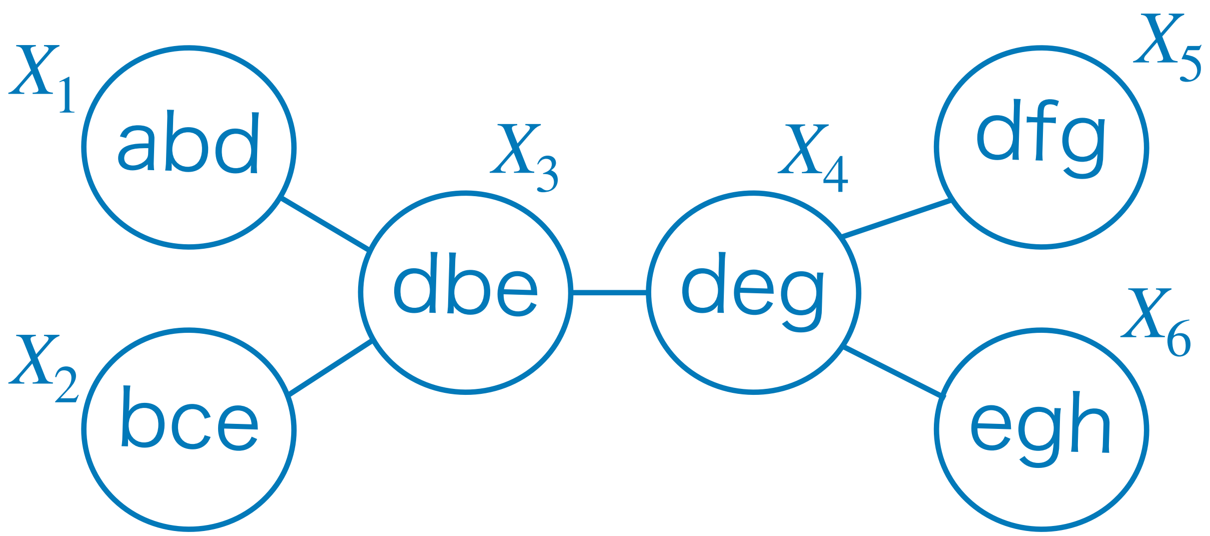
（T2） 全ての辺 $uv \in E(G)$ に対して，
ノード $t \in V(T)$ が存在して $u, v \in X_t$

（T3） 全ての頂点 $v \in V(G)$ に対して，
v を含むバッグに対応するノードの集合は T で連結

入力グラフ $G = (V, E)$



木分解 $\mathcal{T} = \left(T, \{X_t\}_{t \in V(T)}\right)$



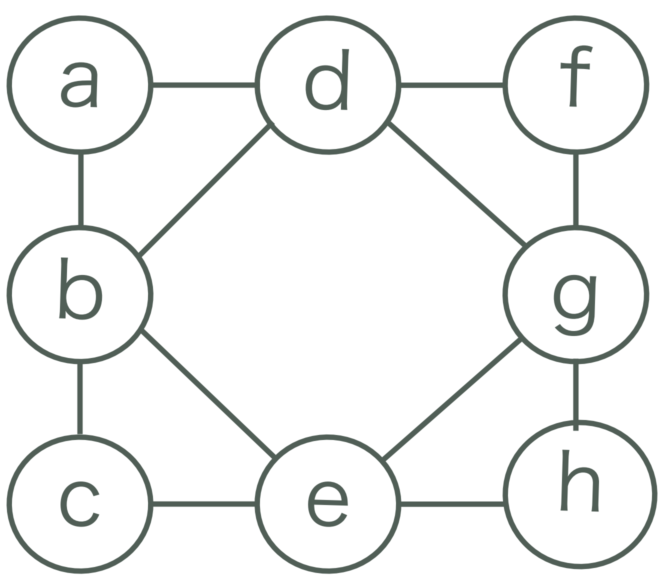
木分解の例（2/2）

（T1） $\bigcup_{t \in V(T)} X_t = V(G)$

（T2） 全ての辺 $uv \in E(G)$ に対して，
ノード $t \in V(T)$ が存在して $u, v \in X_t$

（T3） 全ての頂点 $v \in V(G)$ に対して，
v を含むバッグに対応するノードの集合は T で連結

入力グラフ $G = (V, E)$



木分解 $\mathscr{T} = \left(T, \{X_t\}_{t \in V(T)}\right)$

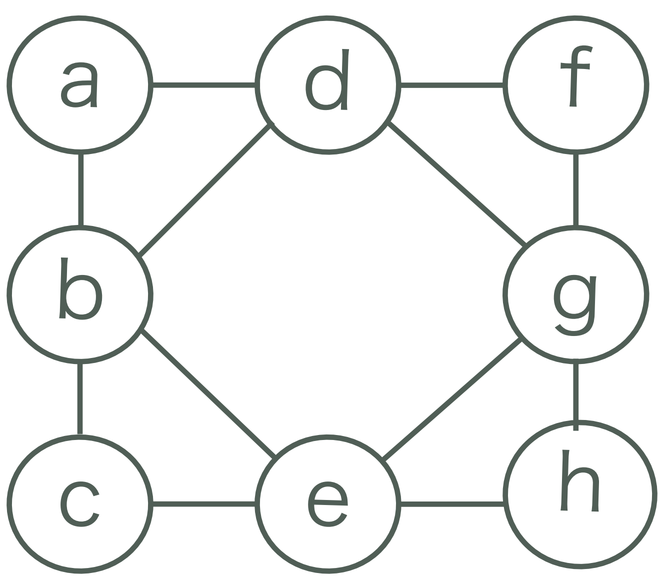
木分解<u>ではない</u>例（1/2）

（T1） $\bigcup_{t \in V(T)} X_t = V(G)$

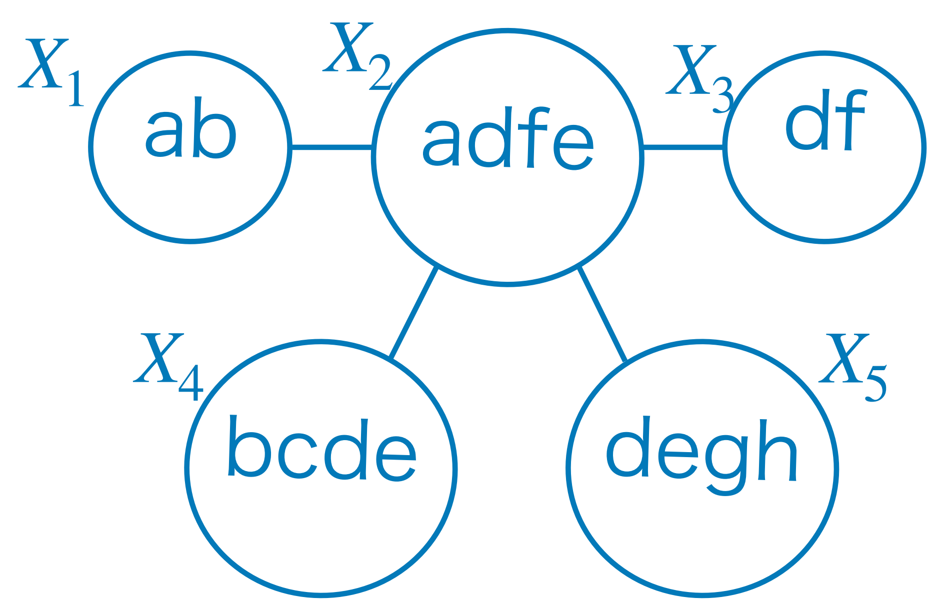
（T2） 全ての辺 $uv \in E(G)$ に対して,
ノード $t \in V(T)$ が存在して $u, v \in X_t$

（T3） 全ての頂点 $v \in V(G)$ に対して,
v を含むバッグに対応するノードの集合は T で連結

入力グラフ $G = (V, E)$



木分解 $\mathcal{T} = \left(T, \{X_t\}_{t \in V(T)}\right)$



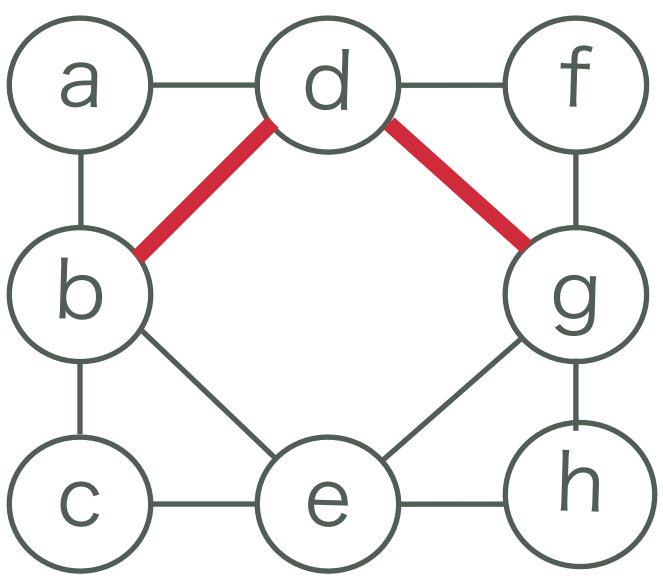
木分解ではない例（1/2）

（T1） $\bigcup_{t \in V(T)} X_t = V(G)$

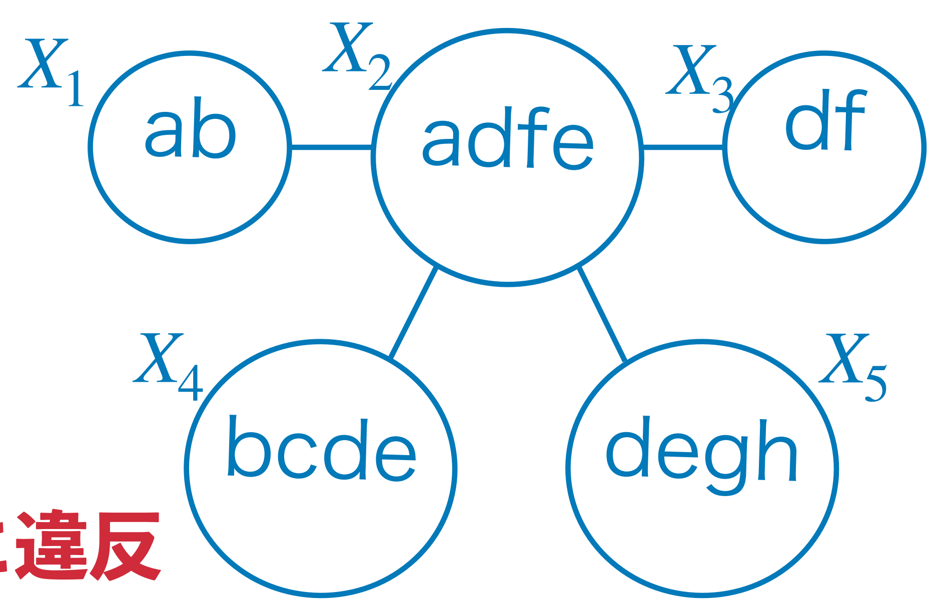
（T2） 全ての辺 $uv \in E(G)$ に対して，
ノード $t \in V(T)$ が存在して $u, v \in X_t$

（T3） 全ての頂点 $v \in V(G)$ に対して，
v を含むバッグに対応するノードの集合は T で連結

入力グラフ $G = (V, E)$



木分解 $\mathscr{T} = \left(T, \{X_t\}_{t \in V(T)}\right)$



（T2）に違反

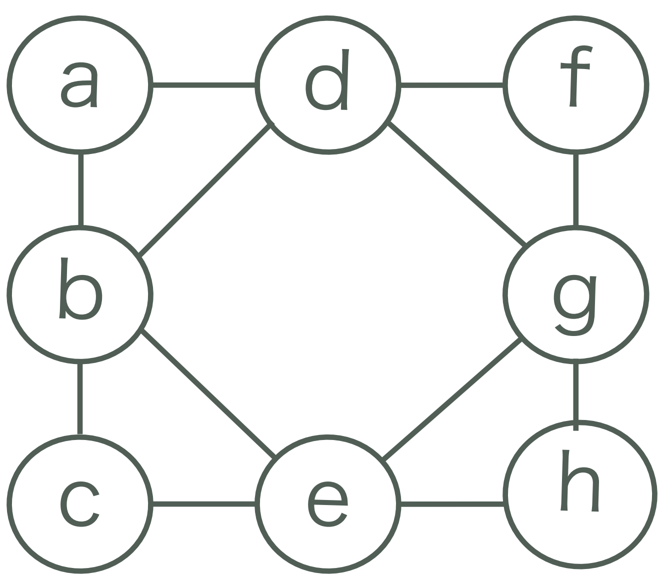
木分解ではない例（2/2）

（T1） $\bigcup_{t \in V(T)} X_t = V(G)$

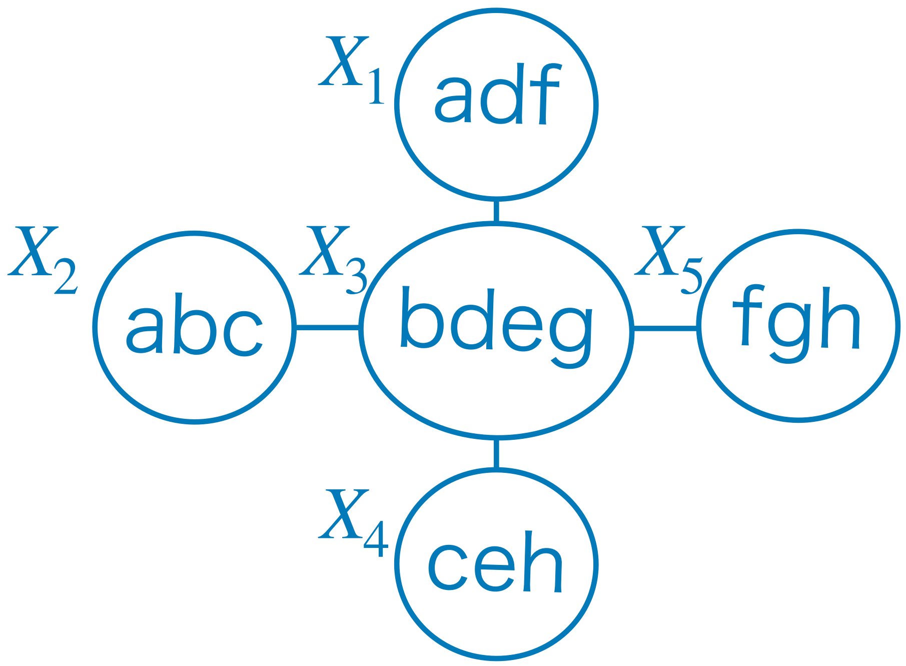
（T2） 全ての辺 $uv \in E(G)$ に対して，
ノード $t \in V(T)$ が存在して $u, v \in X_t$

（T3） 全ての頂点 $v \in V(G)$ に対して，
v を含むバッグに対応するノードの集合は T で連結

入力グラフ $G = (V, E)$



木分解 $\mathcal{T} = \left(T, \{X_t\}_{t \in V(T)}\right)$



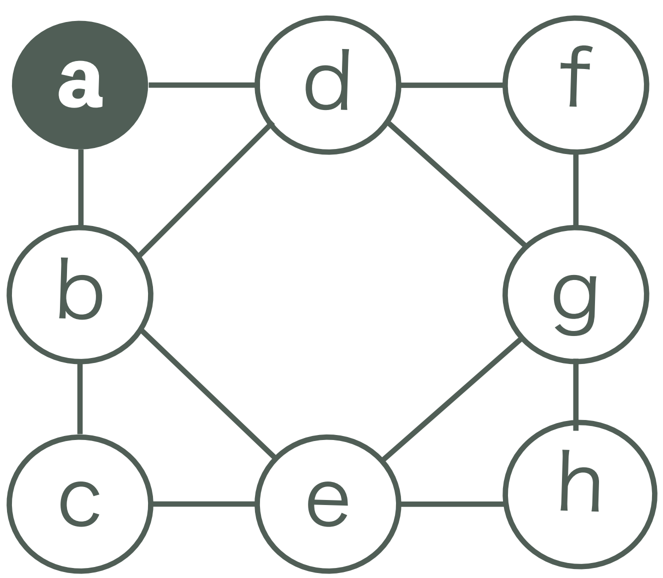
木分解ではない例（2/2）

（T1） $\bigcup_{t \in V(T)} X_t = V(G)$

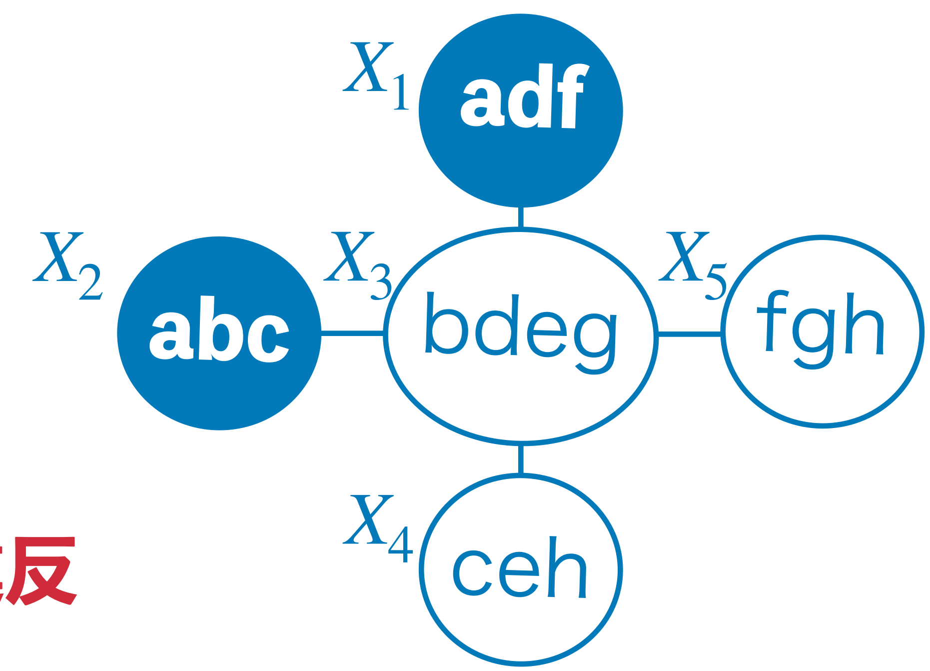
（T2） 全ての辺 $uv \in E(G)$ に対して，
ノード $t \in V(T)$ が存在して $u, v \in X_t$

（T3） 全ての頂点 $v \in V(G)$ に対して，
v を含むバッグに対応するノードの集合は T で連結

入力グラフ $G = (V, E)$



木分解 $\mathscr{T} = \left(T, \{X_t\}_{t \in V(T)}\right)$



（T3）に違反

木幅の定義

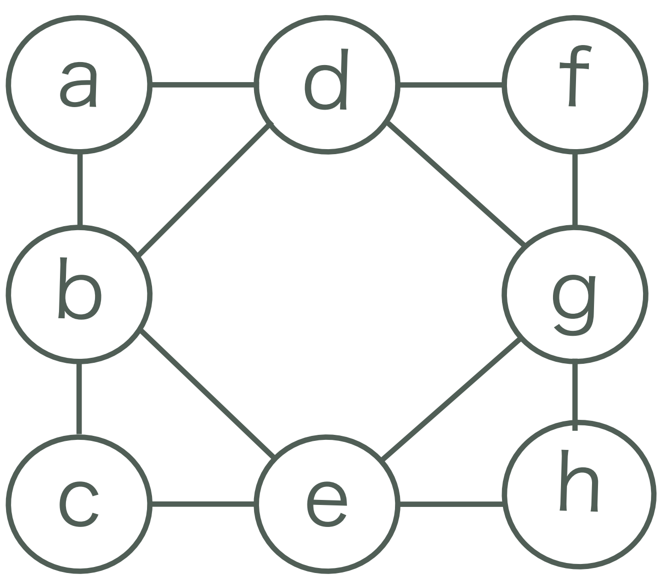
木分解 $\mathscr{T}$ の**幅（width）**とは,

最大のバッグサイズから，１引いたもの

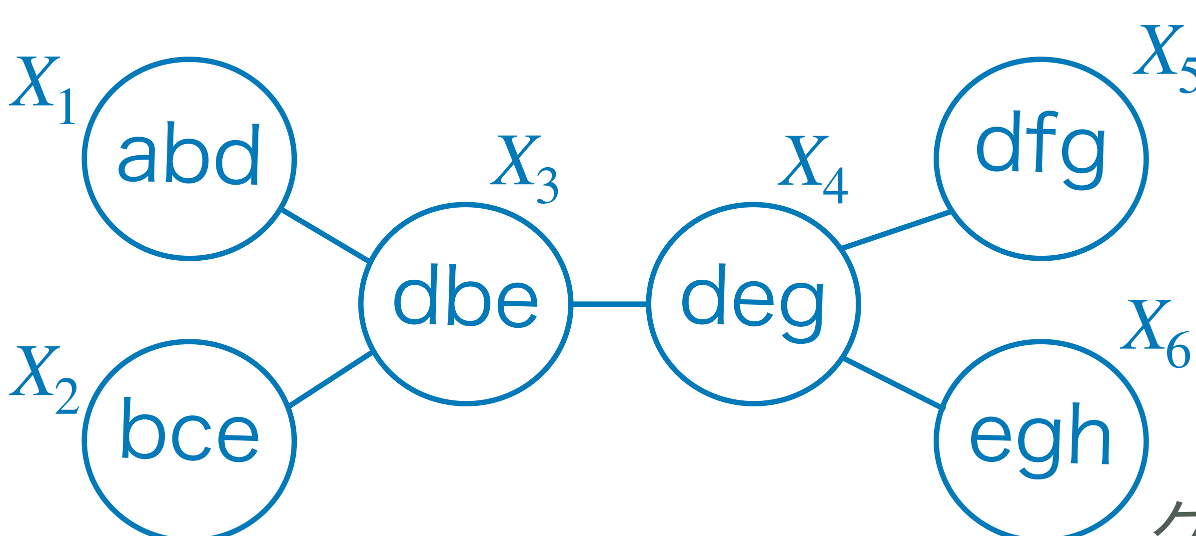
グラフ G の**木幅（tree-width）**とは,

全ての木分解を考えたときの最小の幅

入力グラフ $G = (V, E)$



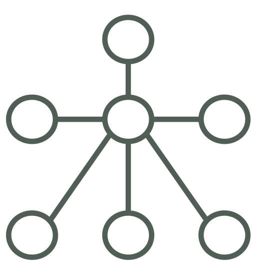
木分解 $\mathscr{T} = \left(T, \{X_t\}_{t \in V(T)}\right)$



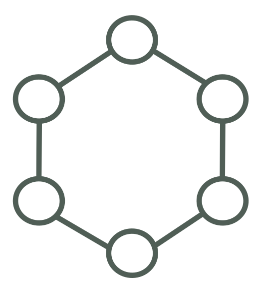
グラフ G の木幅は 2

木幅とは（再掲）

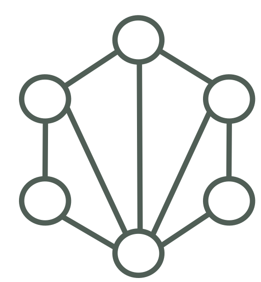
- **木幅（tree-width）：**
  無向グラフに対して定義される不変量の一つ
  - 大雑把にいうと，グラフの<u>**木っぽさ**</u>を表す指標
  - 木幅が小さいほどグラフは<u>**木っぽい**</u>



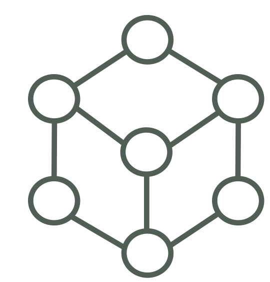
木：tw $= 1$



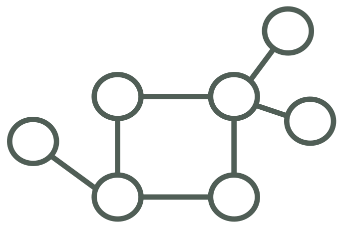
サイクル：tw $= 2$



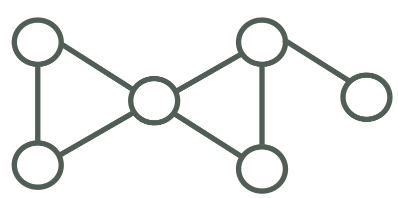
外平面グラフ：tw ≤ 2



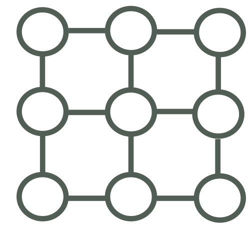
平面グラフ：tw $= O(\sqrt{n})$



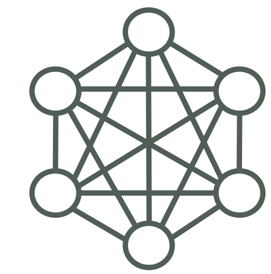
擬似木：tw $= 2$



カクタス木：tw ≤ 2

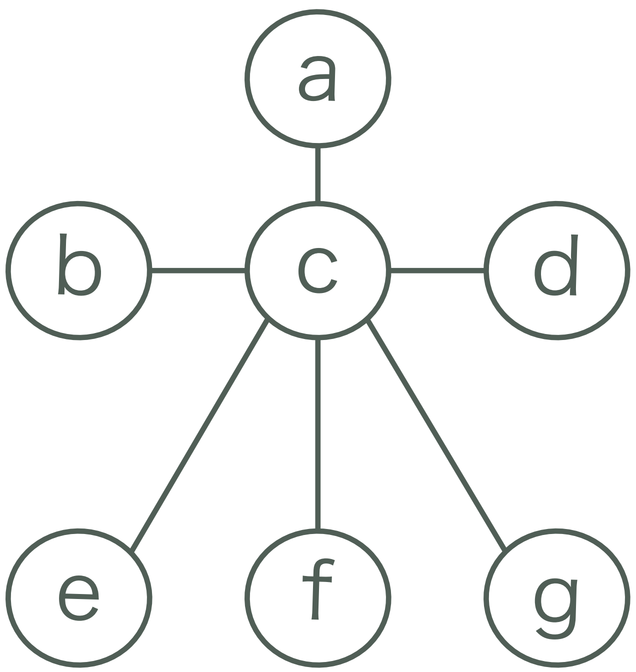


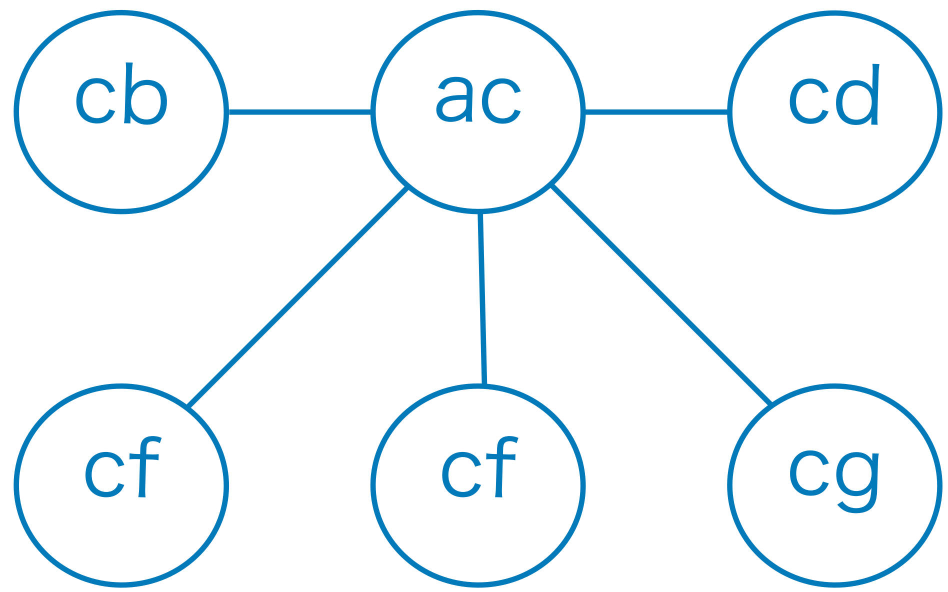
グリッド：tw $= \sqrt{n}$



完全グラフ：tw $= n - 1$

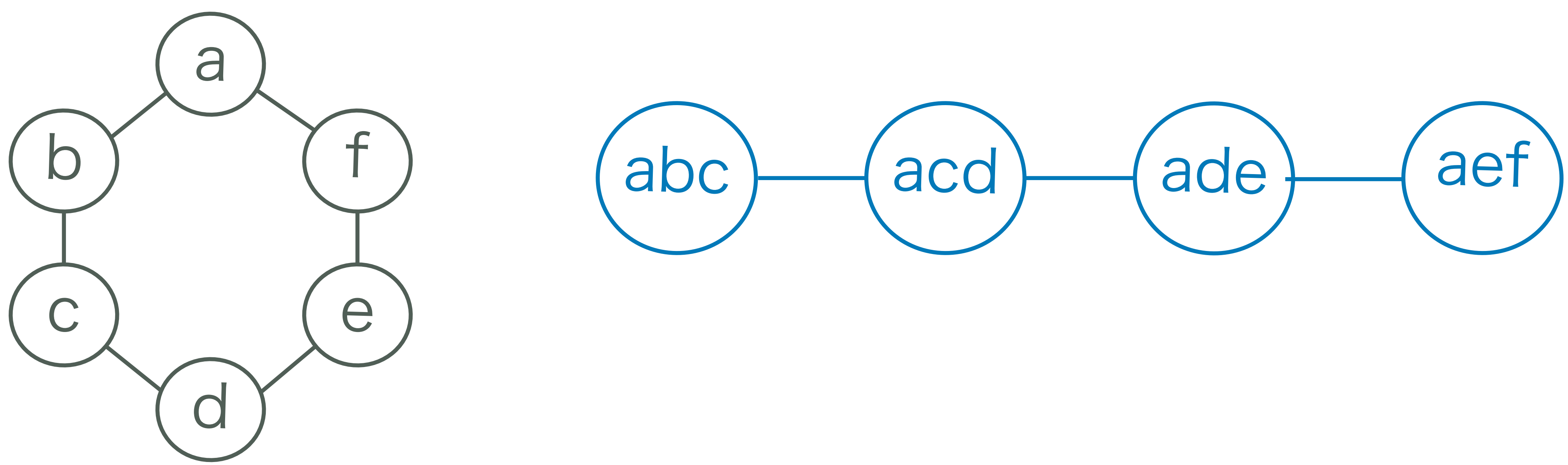
様々なグラフの木幅（1/2）





木：tw = 1

様々なグラフの木幅（2/2）



サイクル：tw = 2

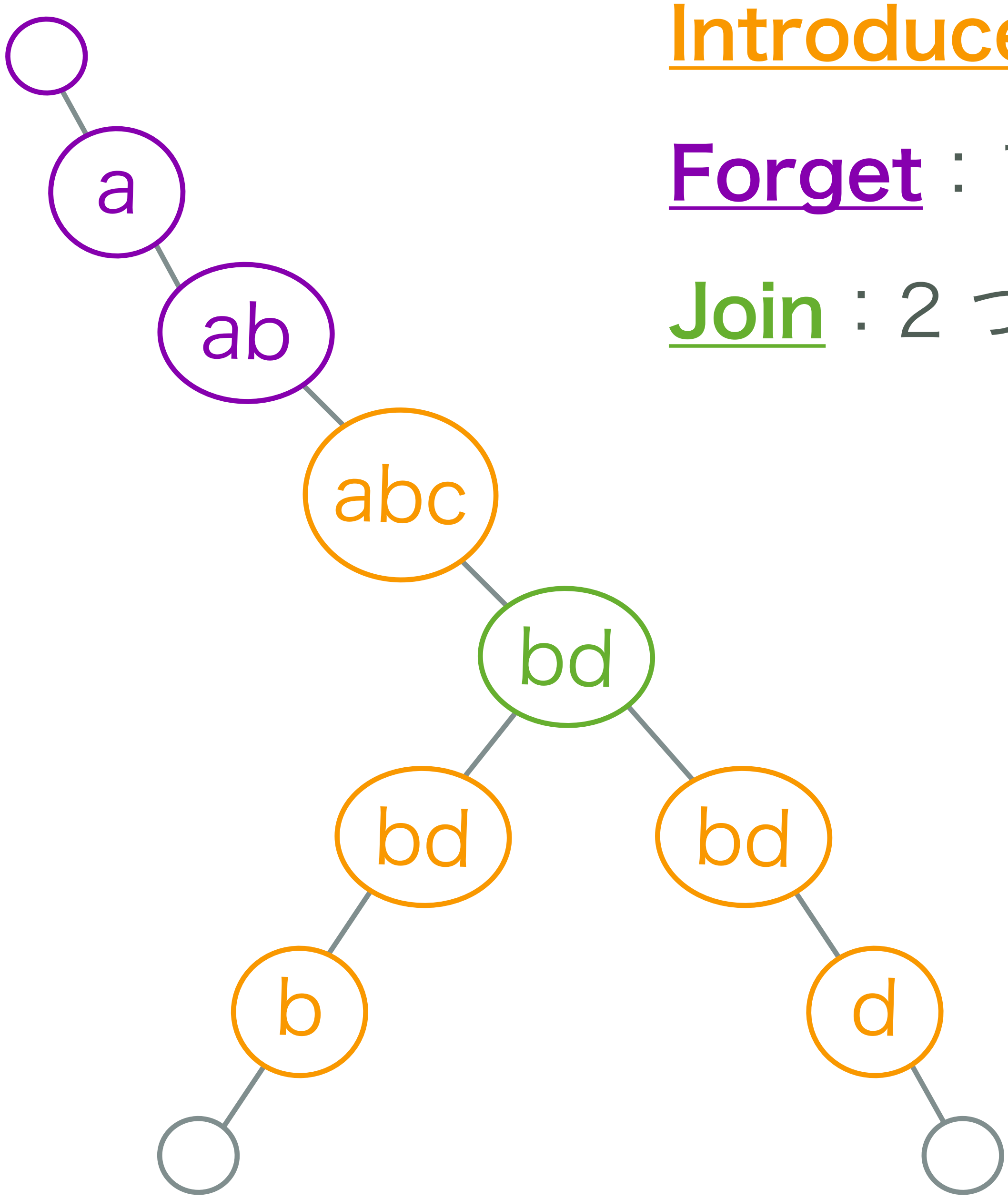
素敵な木分解

木分解 $\mathscr{T}$ が以下の二つの性質を満たすとき,
木分解 $\mathscr{T}$ は**素敵な木分解（nice tree-decoposition）**であるという

- $X_r = \emptyset, X_l = \emptyset$（根と葉に対応するバッグは空集合）
- 葉以外の全てのノード t は，以下のいずれかのタイプに属する
  - **Introduce**：1 つの子 t' を持ち，$X_t = X_{t'} \cup \{v\}$
  - **Forget**：1 つの子 t' を持ち，$X_t = X_{t'} \backslash \{v\}$
  - **Join**：2 つの子 t'_1, t'_2 を持ち，$X_t = X_{t'_1} = X_{t'_2}$

注：$\mathscr{T} = \left(T, \{X_t\}_{t \in V(T)}\right)$ の T は，根付き木とする

素敵な木分解の例



Introduce：1 つの子 t' を持ち，$X_t = X_{t'} \cup \{v\}$

Forget：1 つの子 t' を持ち，$X_t = X_{t'} \setminus \{v\}$

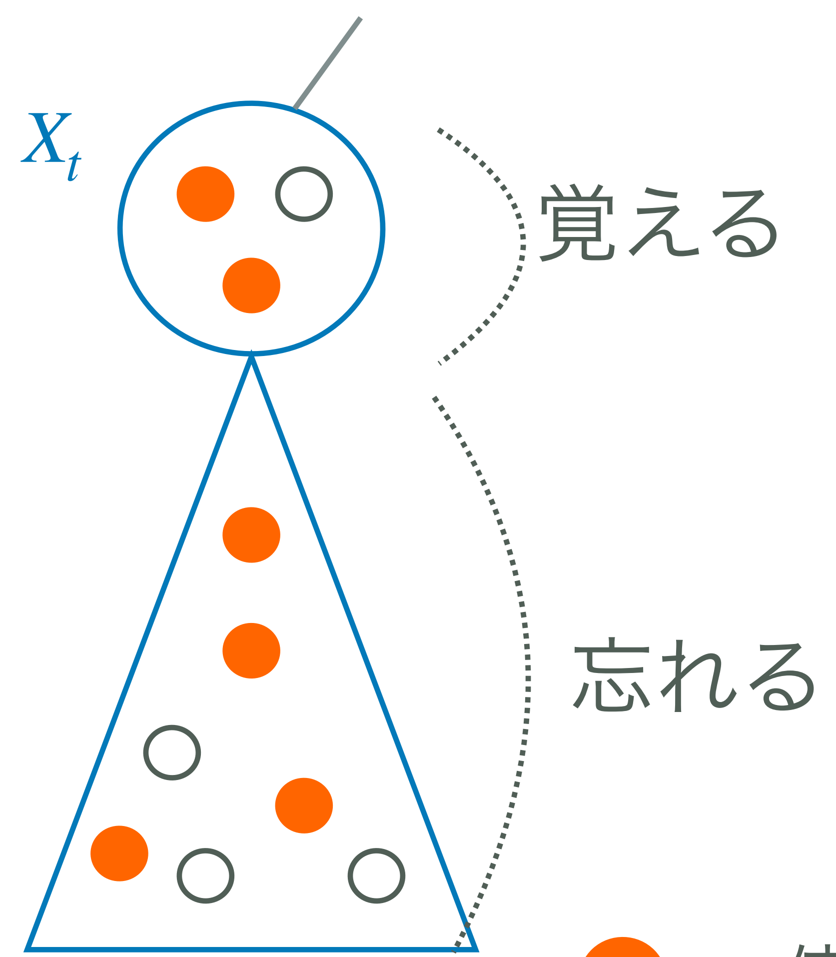
Join：2 つの子 t'_1, t'_2 を持ち，$X_t = X_{t'_1} = X_{t'_2}$

幅 w の木分解から，
幅 w の素敵な木分解を
$O(w^2 n)$ 時間で得ることができる

素敵な木分解上の動的計画法

- $dp[t, S] :=$ 各ノード t と部分集合 $S \subseteq X_t$ について，
  $I \cap X_t = S$ であるような独立点集合 $I \subseteq V_t$ の最大重み和

V_t：t を根とする根付き木に含まれる頂点



- **求める解**：$dp[r, \emptyset]$
- **初期条件**：$dp[l, \emptyset] = 0$（l は葉）

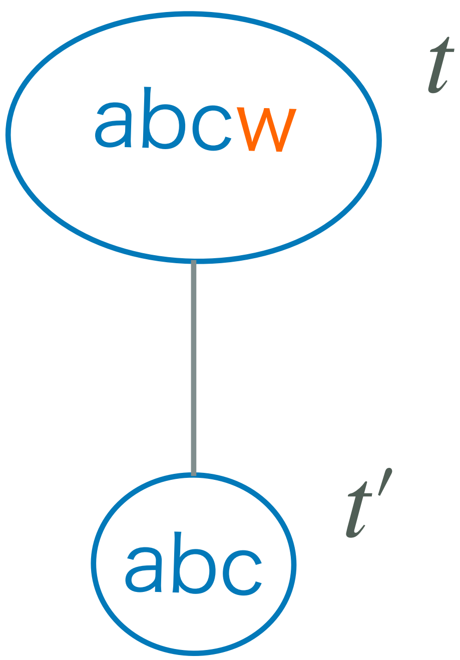
ボトムアップに（葉から順に）
DP テーブルを更新していく

●：使う頂点　○：使わない頂点

動的計画法の更新式（Introduce）

t が Introduce node のとき

$$dp[t,S] = \begin{cases} dp[t',S] & \text{if } v \notin S, \\ dp[t', S \setminus \{v\}] + w(v) & \text{otherwise}. \end{cases}$$

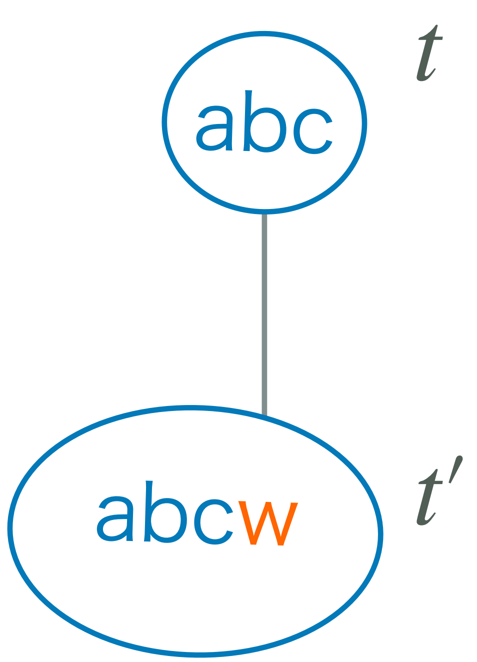


更新式の気持ち：
頂点 w を加えるか，加えないかで場合分け

動的計画法の更新式（Forget）

t が Forget node のとき

$$dp[t, S] = \max\left\{dp[t', S], dp[t', S \cup \{w\}]\right\}$$



更新式の気持ち：

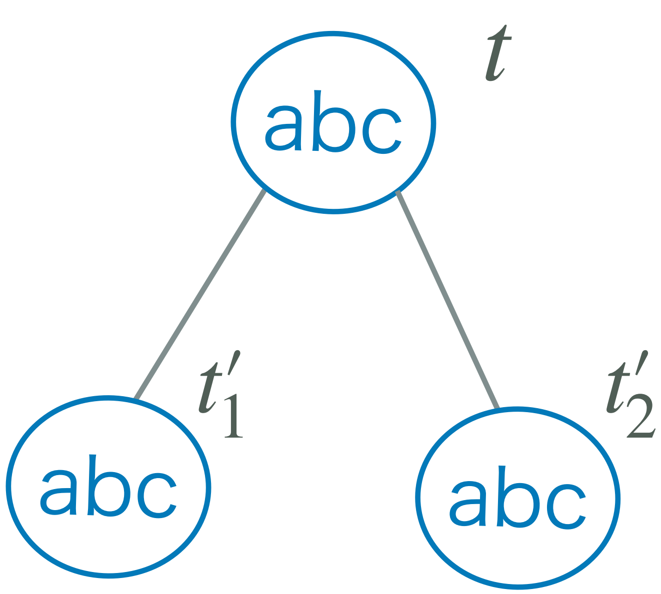
新たに頂点 w を忘れるか,

すでに忘れていたかの場合分け

動的計画法の更新式（Join）

t が Join node のとき

$$dp[t, S] = dp[t'_1, S] + dp[t'_2, S] - w(S)$$



更新式の気持ち：

重複で 2 回足している $w(S)$ を引く

アルゴリズムの計算量

- DP テーブルのサイズは，各ノードで高々 2^{w+1} であり，
  テーブルの更新は $O(w)$ でできるため，
  全体で $O(2^w wn)$ となる

- 以上より，以下の定理を得る

主定理：

入力として重み付き無向グラフ $G=(V,E), w: V \to \mathbb{R}$ と，

バッグのサイズが高々 w の木分解 $\mathcal{T} = \left(T, \{X_t\}_{t \in V(T)}\right)$ が与えられた

とき，重み付き最大独立点集合問題を $O(2^w wn)$ 時間で解く

まとめ

- 入力グラフの木幅が定数で抑えられるとき,
  多項式時間で動作する動的計画法が設計できることがある

- 本スライドでは，重み付き最大独立点集合を紹介した
  - 木幅を w として，$O(2^w wn)$ 時間で解けることを示した

- 他にも様々な問題が,
  同様の動的計画法で解けることが知られている
  - 最小頂点被覆問題：$2^w \cdot k^{O(1)} \cdot n$ 時間
  - 支配集合問題：$4^w \cdot w^{O(1)} \cdot n$ 時間
  - 最大カット問題：$2^w \cdot w^{O(1)}$ 時間
  - など