Maximum Weighted Independent Set

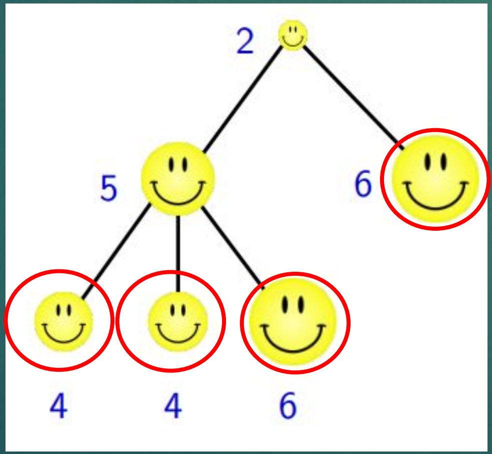
MADE BY, NAZMUL ISLAM

ROLL- 1605088

Table of Content

The Party Problem

Problem: Invite some colleagues for a party
Maximize: The total fun factor of the invited people.
Constraint: Everyone should be having fun.
Do not invite a colleague and his direct boss at the same time!



Input: A tree with weights on the vertices.
Task: Find an independent set of maximum weight.

Solving the Party Problem

- **Subproblems:**
  - **T_v** : The subtree rooted at v.
  - **A[v]:** maximum weight of an independent set in T_v that contains v
  - **B[v]:** maximum weight of an independent set in T_v that does not contain v
- **Goal:** Determine A[r] for the root r.

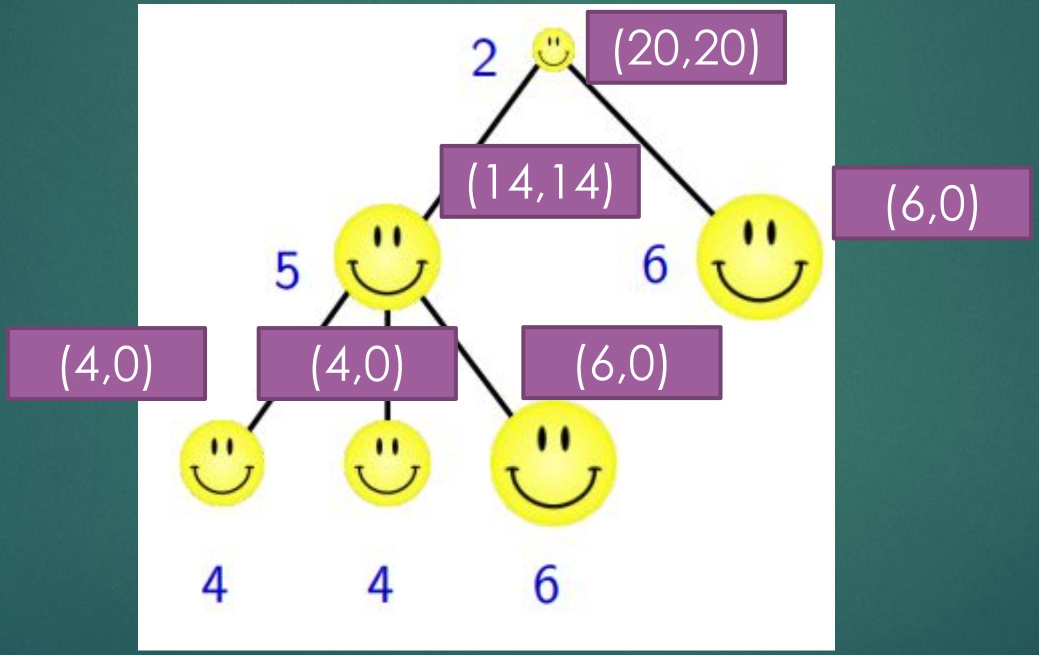
Solving the Party Problem

- **Recurrence:**
  Assume $v_1, v_2, \ldots, v_k$ are the children of v. Use the recurrence relations

$$B[v] = \sum_{i=1}^{k} A[v_i]$$
$$A[v] = \max\{B[v]\ ,\ w(v) + \sum_{i=1}^{k} B[v_i]\}$$

- The values A[v] and B[v] can be calculated in a bottom-up order.

Solution

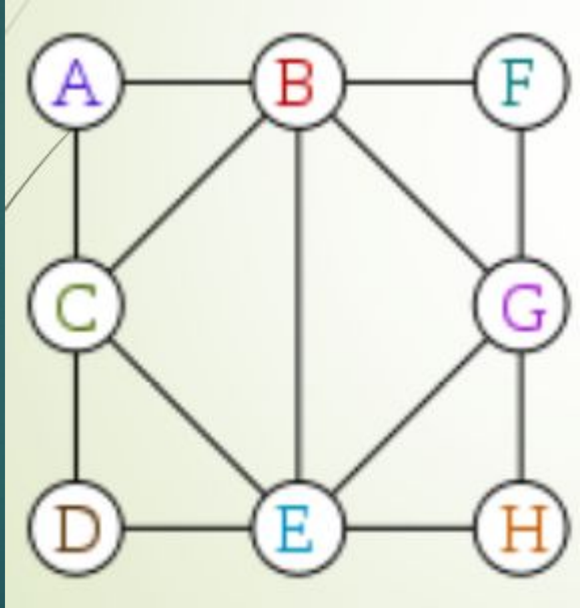


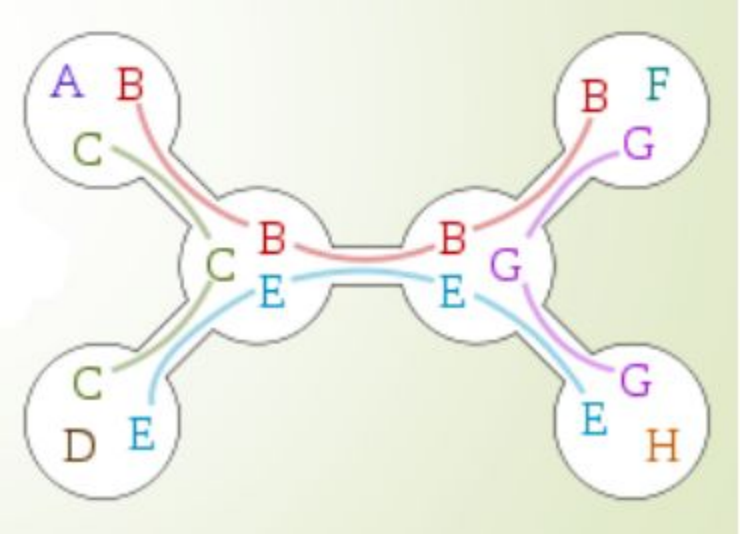
Treewidth of a Graph

- The **treewidth** of an undirected graph is a number associated with the graph
- Treewidth captures how similar a graph is to be a tree
- Treewidth is commonly used as a parameter in the parameterized complexity analysis of graph algorithms.
- In this slide, we focus on connections to the idea of dynamic programming on the structure of a graph.

Tree Decomposition

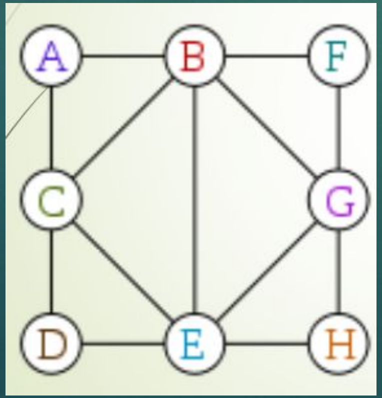
- A **tree decomposition** is a mapping of a graph into a tree that can be used to define the treewidth of the graph.

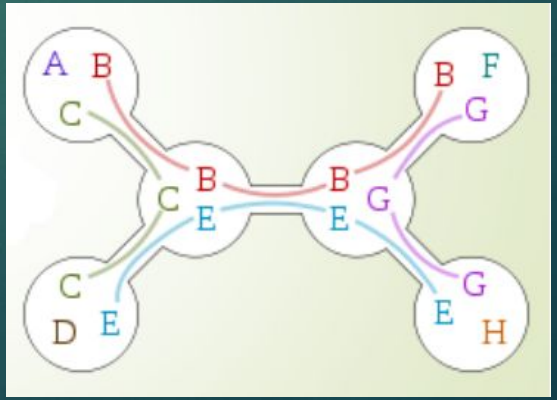




Tree Decomposition

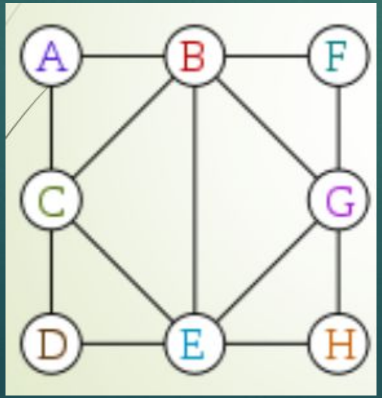
- Definition: a tree decomposition of a graph G is a pair $\mathcal{T} = (T, Xt\ t{\in}V\ T)$, where T is a tree whose every node t is assigned a vertex subset $Xt \subseteq V(G)$.
- The following three conditions hold:
  - $X_t = V(G)$
  - For every $uv \in E(G)$, there exists a node t of T such that bag X_t contains both u and v
  - For every u ϵ V(G), the set $T_u = \{t \in V(T): u \in X_t\}$, induces a connected subtree of T.

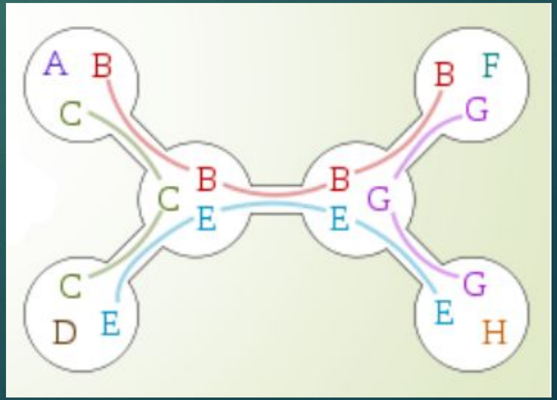




Treewidth by a Tree Decomposition

- After we defined what a tree decomposition is , we can define the treewidth of a graph
- The width of tree decomposition equals $\max_{t \in V(T)} |X_t| - 1$
- The treewidth of a graph G, denoted by $tw(G)$, is the minimum possible
width of a tree decomposition of G



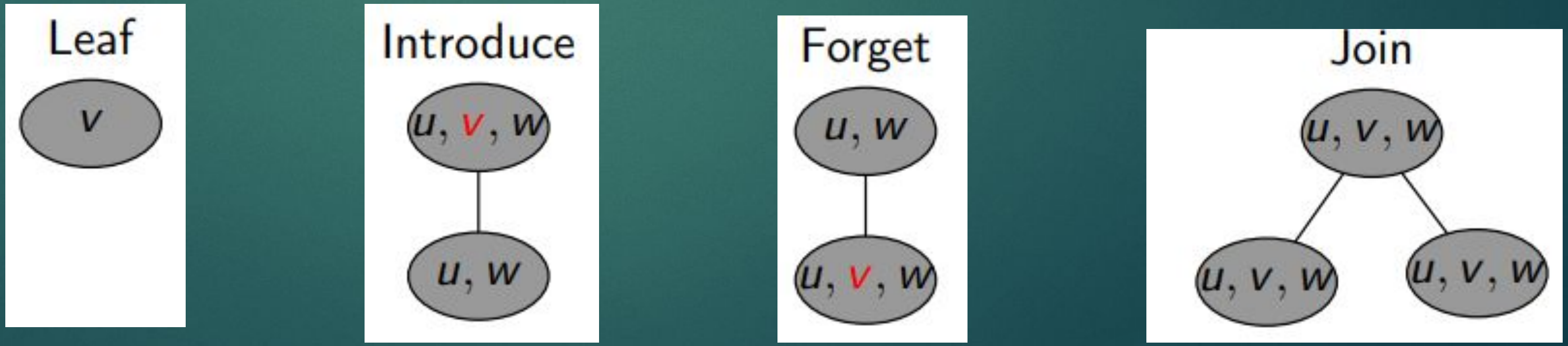


Maximum Weighted Independent Set

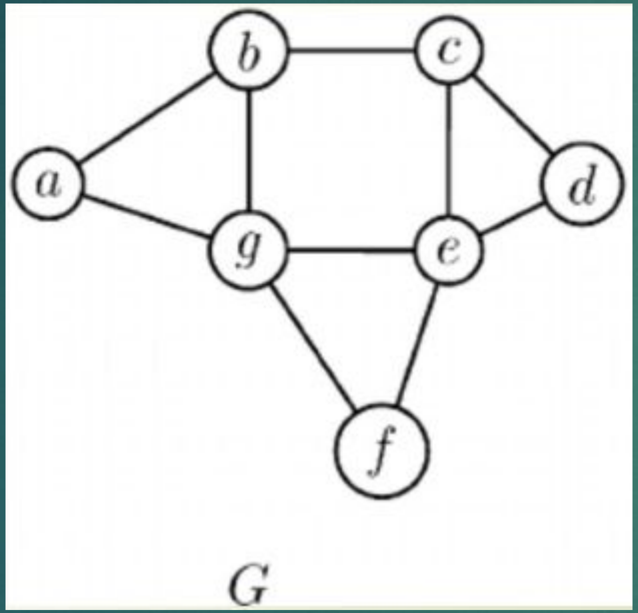
- Problem: Given a tree decomposition of treewidth w, we need to find the maximum weighted independent set .
- The maximum weighted independent set is known to be NP-hard
- Therefore, it is unlikely that there exists an efficient algorithm for solving it.
- However , we will now see a dynamic-programming-based algorithm that solves it efficiently on graphs of bounded treewidth.

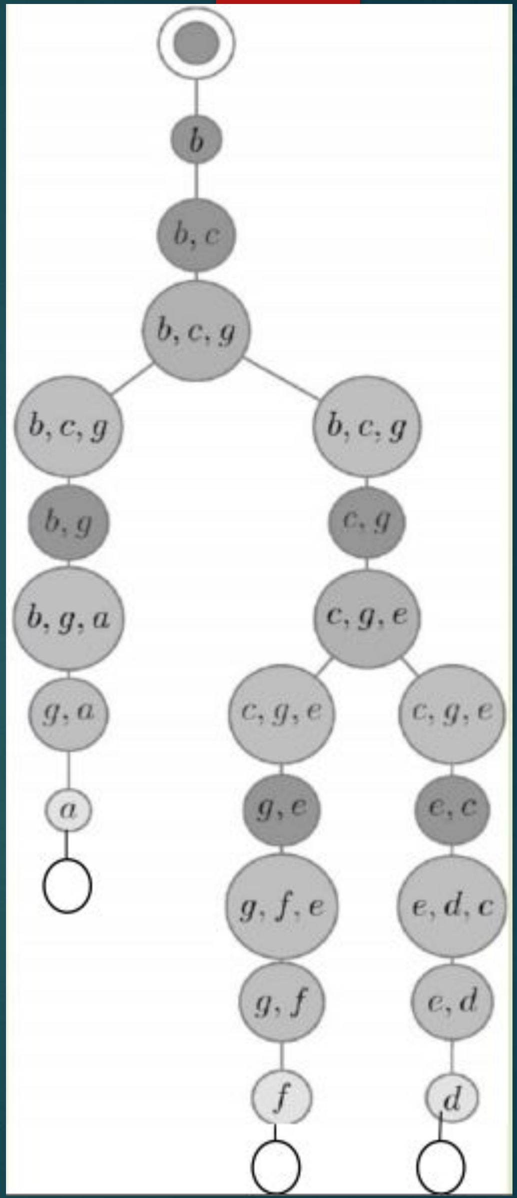
Nice tree decompositions

- Definition: A rooted tree decomposition is nice if every node x is one of the following 4 types:
  - **Leaf:** No children, $|B_x|=1$
  - **Introduce:** 1 child y with $B_x=B_y \cup \{v\}$ for some vertex v
  - **Forget:** 1 child y with $B_x=B_y \setminus \{v\}$ for some vertex v
  - **Join:** 2 children y_1,y_2 with $B_x=B_{y1}=B_{y2}$



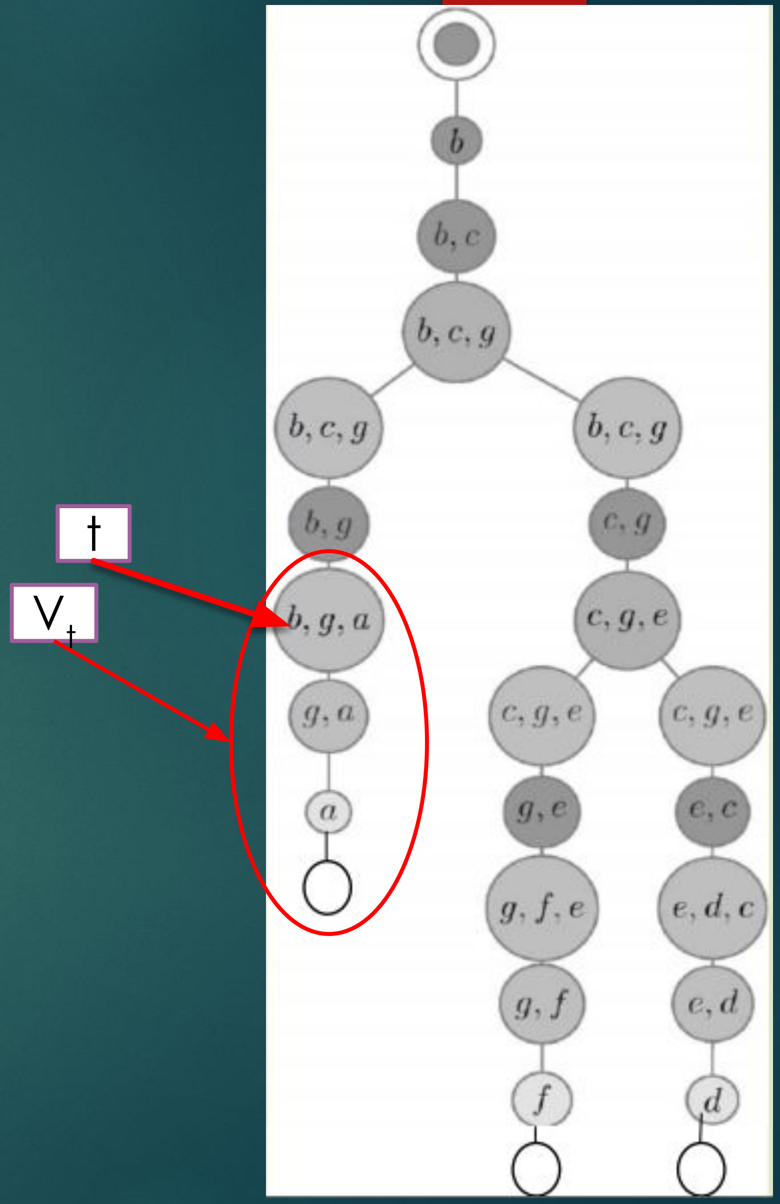
Nice tree decompositions





Weighted Independent Set

- Recall that T is rooted at some node r
- For a node t of T, let V_t be the union of all the bags present in the subtree of T rooted at t, including X_t.

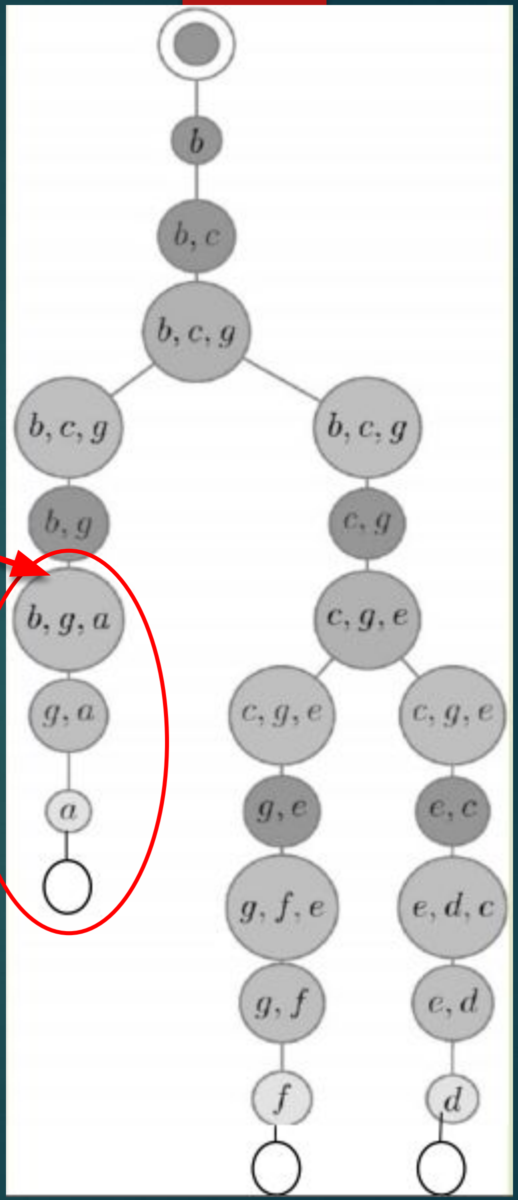


Weighted Independent Set

- Among independent sets I satisfying $I \cap X_t = S$ for some fixed S, all the maximum-weight solutions have exactly the same weight of the part contained in V_t
- For every node t and every S subset X_t, define the following value:
  - $C[t,S]$ = The maximum weight of an independent set I subset V_x with $I \cap X_t = S$

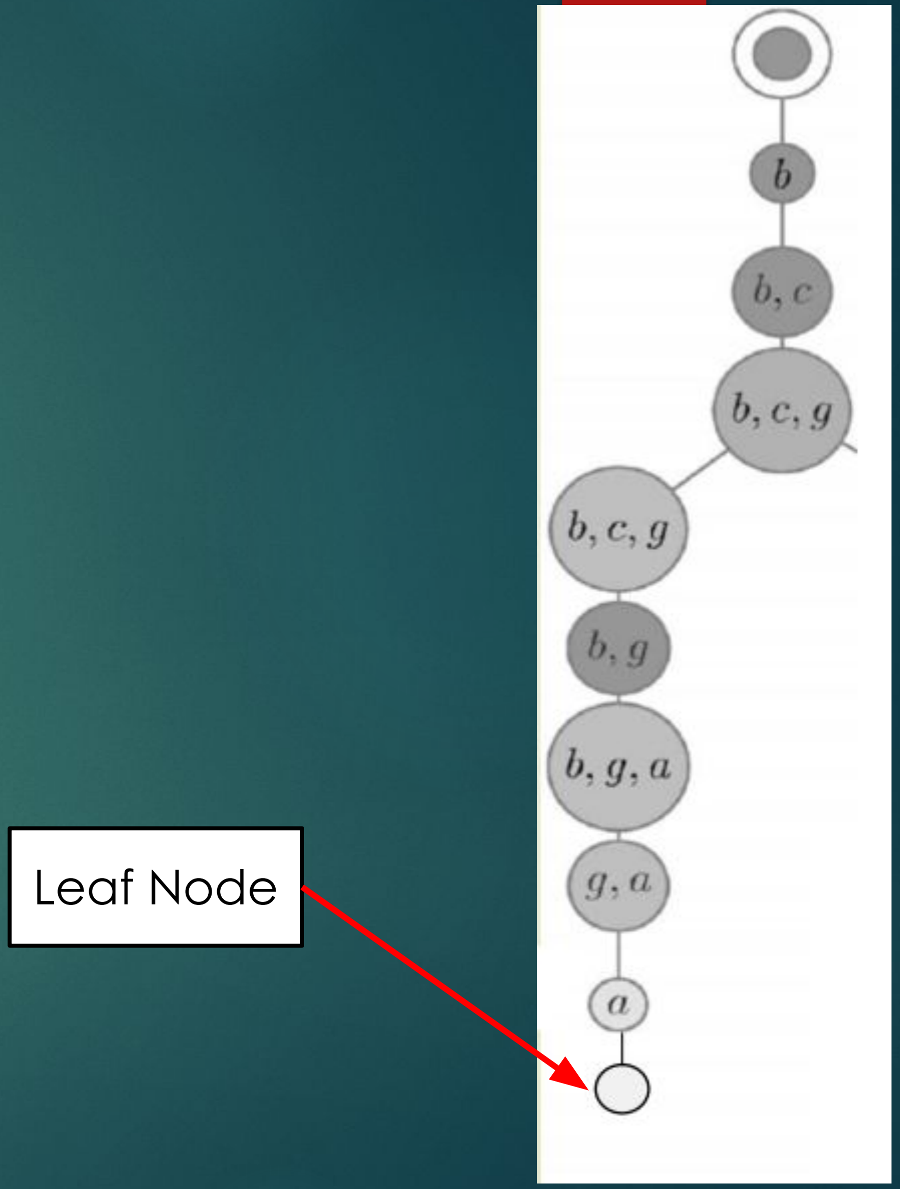
t

V_t



Weighted Independent Set

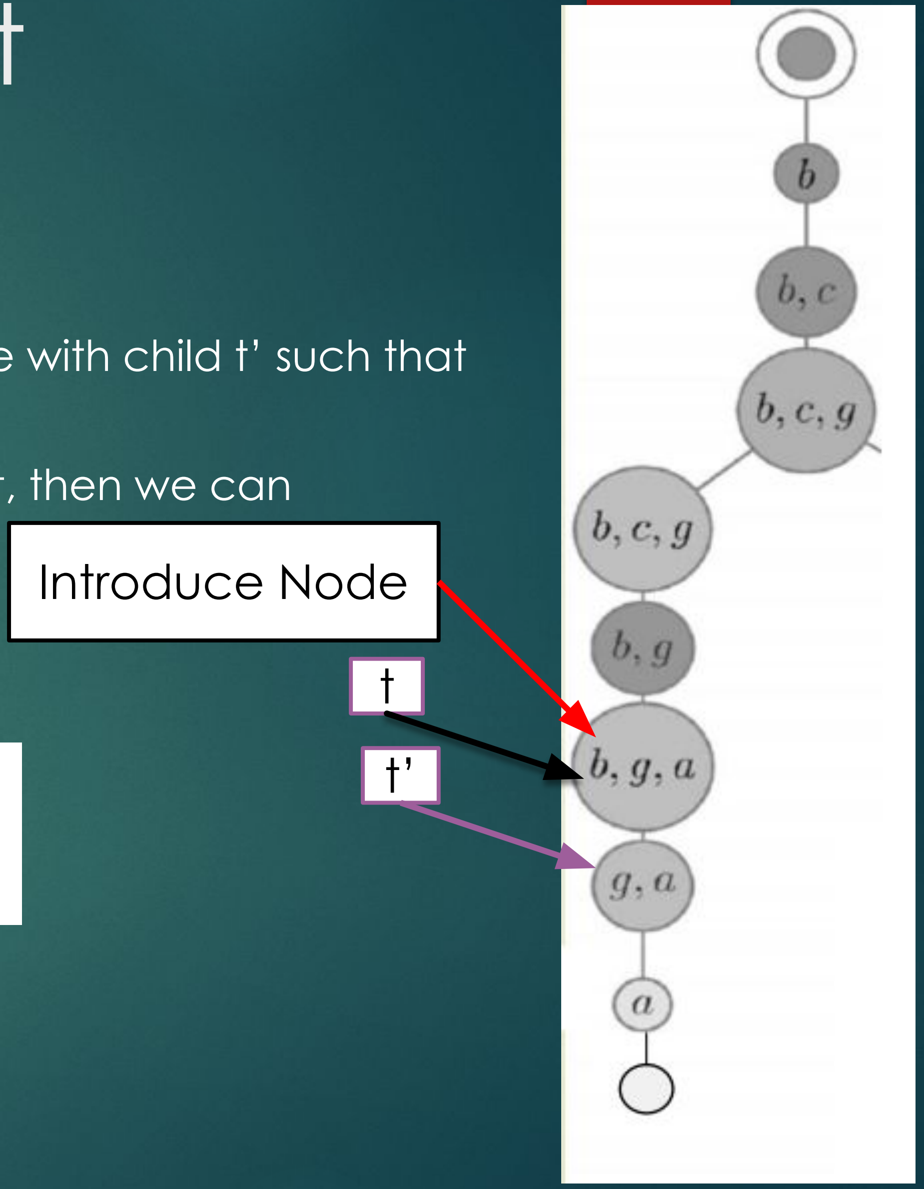
- **Leaf node:** If t is a leaf node, then we have only one value c[t,ø] = 0



Weighted Independent Set

- **Introduce node:** Suppose t is an introduce node with child t’ such that $X_t = X_{t'} \cup \{v\}$
- Let S be any subset of X_t. If S is not independent, then we can immediately put $c[t,S] = -\infty$;
- Otherwise the following formula holds:

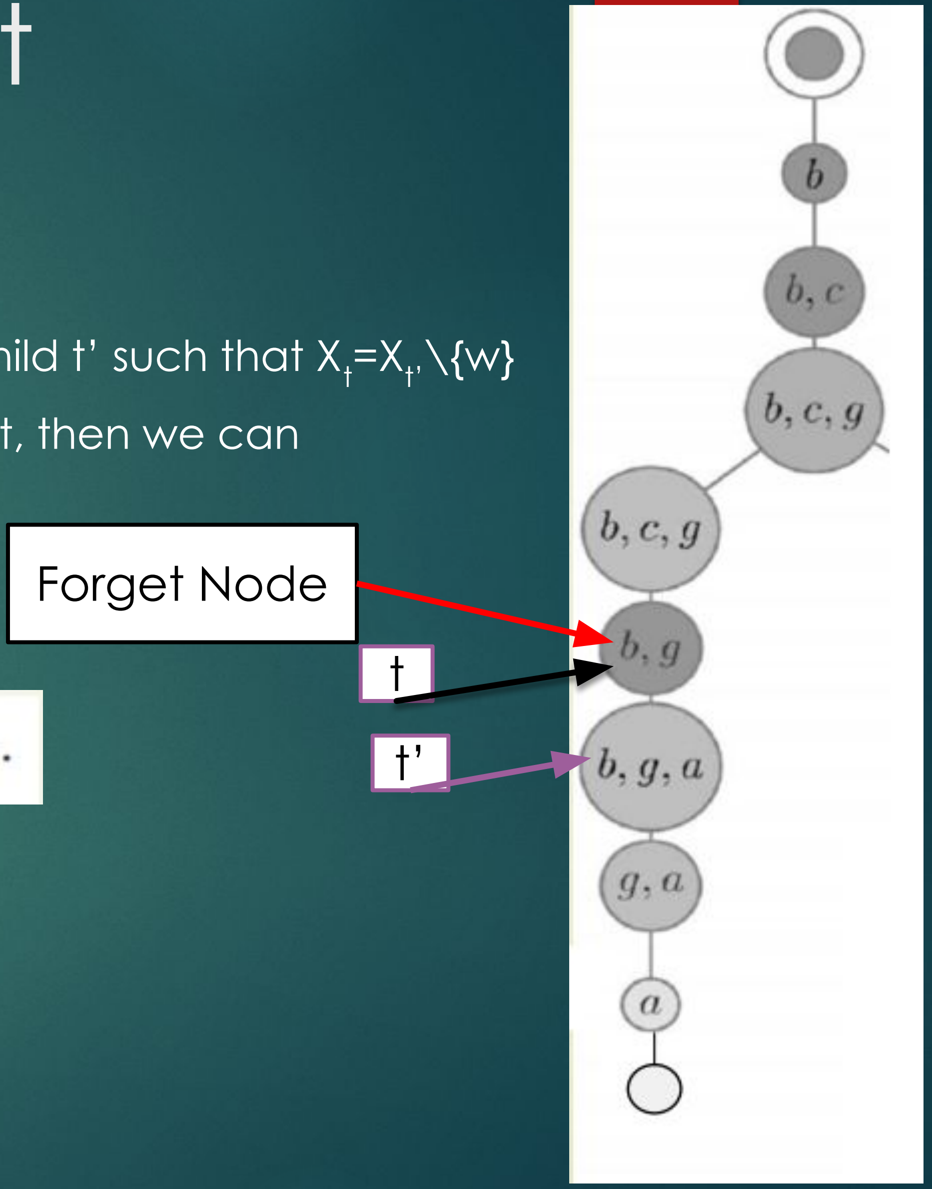
$$c[t,S] = \begin{cases} c[t',S] & \text{if } v \notin S; \\ c[t', S \setminus \{v\}] + \mathbf{w}(v) & \text{otherwise.} \end{cases}$$



Weighted Independent Set

- **Forget node:** Suppose t is a forget node with child t' such that $X_t = X_{t'} \setminus \{w\}$
- Let S be any subset of X_t. If S is not independent, then we can immediately put $c[t,S] = -\infty$;
- Otherwise the following formula holds:

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



Weighted Independent Set

Join Node

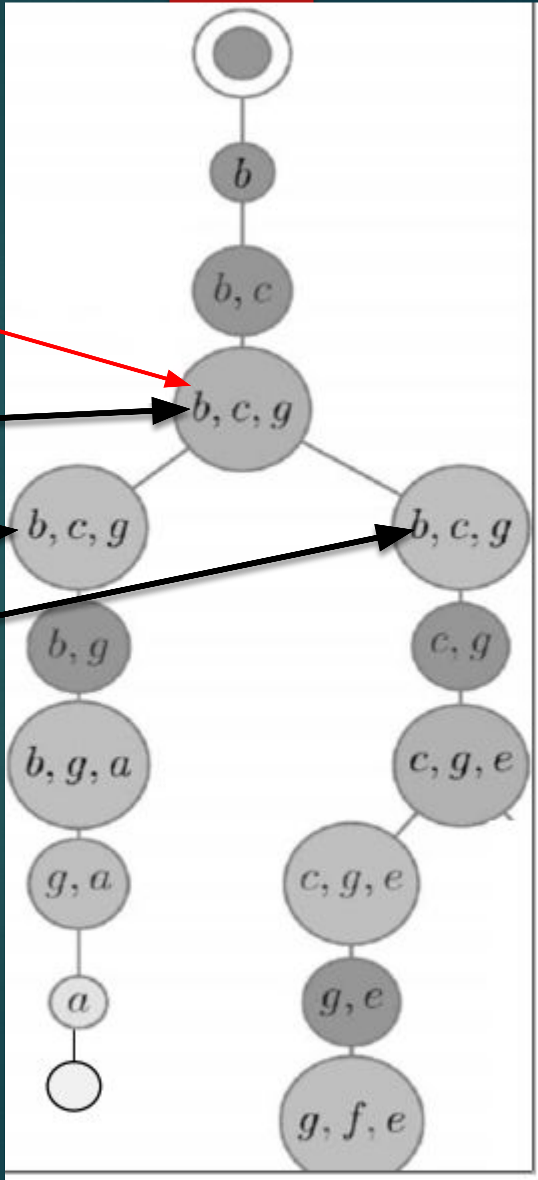
- **Join node:** Suppose t is a join node with children t_1,t_2 such that X_t=X_{t1}=X_{t2}
- Let S be any subset of X_t. If S is not independent, then we can immediately put c[t,S] = -∞;
- Otherwise the following formula holds:

$$c[t, S] = c[t_1, S] + c[t_2, S] - \mathbf{w}(S).$$

t

t_1

t_2



Thank You for Listening