6. $\boxed{\text{10 marks}}$ **(Take Home Question.)** In the SMALL DISCONNECTED LARGE SUBGRAPH problem we are given a graph G on n vertices, integers k and q, and the goal is to check if there exists $X \subseteq V(G)$ of size at most k, such that $G - X$ has at least two connected components each with at least q vertices. Design a deterministic $2^{O(k+q)} \cdot n^{O(1)}$-time algorithm for the above problem.
Hint: Min-cut-Max-flow

Note: We use the minflow computing algo as a black-box → in poly time in number of vertices, say using Edmonds–Karp. or push-relabel

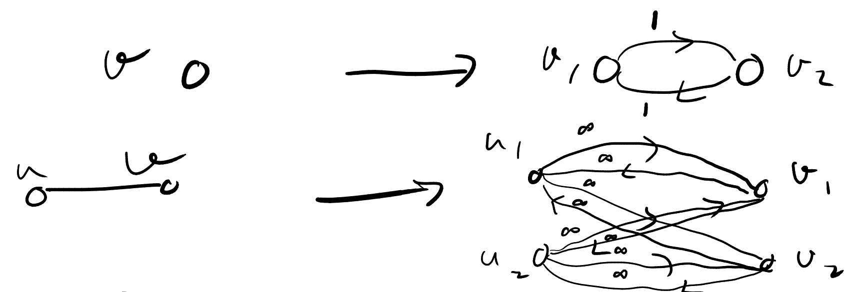
Min s-t-vertex cut: Given graph G, distinct non-adjacent vertices $s, t \in V(G)$, find minimum set $X \subseteq V(G) \setminus \{s,t\}$ s.t. $G \setminus X$ has s and t in two distinct components in poly time

Lemma 1: We can find min. s-t-vertex cut in poly time.

Proof: Reduction from min. s-t-vertex cut to mincut/maxflow in digraphs
Let the digraph be D.

Type I edge | For every vertex $v \in V(G) \setminus \{s,t\}$, introduce 2 vertices v_1 & v_2 in digraph D with a weight 1 bidirectional edge between them.

Type II edge | $\forall\, uv \in E(G)$, construct 8 edges $uv_1, uv_2, u_2v_1, u_2v_2$, $v_1u_1, v_1u_2, v_2u_1, v_2u_2$. [if u, v are s/t, some of these edges will be degenerate] of infinite capacity.



Type I edge

Type II edge

$V(D) = \{v_1, v_2 \;\forall\; v \in V(G) \setminus \{s,t\}\} \cup \{s,t\}$
$E(G)$ is union of type I & II edges.

Claim 1: Let $\exists$ a st vertex cut in the original graph of size k
We claim that in the reduced instance, $\exists$ a flow of size k.

Let the st vertex cut be $X \subseteq V(G) \setminus \{s,t\}$.
Consider the foll cut in $D \rightarrow$ The cut corresponding to separating all the type I edges corresponding to vertices in X
Note that the size of the cut is exactly k.

Thus, $k \geq$ min-cut $=$ maxflow [min cut - maxflow thm]

Claim 2: Let $\exists$ a cut C in the reduced digraph of value k.
We claim that in the original instance, $\exists$ a st vertex cut of size k.

The cut size is atmost k, thus we cannot pick any type II edge
Thus, only Type-I edges cross the cut.
Let X be the vertices in G corresponding to type I edges across the cut.
Clearly, this is a valid st vertex cut in G.

Using Claim 1 & Claim 2, we get that size of min st vertex cut of G $=$ mincut in D $=$ maxflow in D. Thus, lemma 1 is proved.

Defn · Suppose it is a Yes instance of SMALL DISCONNECTED LARGE SUBGRAPH. i.e. $\exists$ some soln X and 2 components F_1, F_2 of at least q vertices, $|X| \le k$. Fix a connected subgraph of F_1 with q vertices, say C_1 and a connected subgraph of F_2 of q vertices, say C_2. $|C_1| = |C_2| = q$. Define a $\boxed{\text{good partition}}$ to be a partition (L, R) of $V(G)$ to be s.t. $X \subseteq L$, and $C_1 \subseteq R$ and $C_2 \subseteq R$.

Clearly a good partition always exists for "Yes" instance.

Lemma 2: Given a good partition (L, R) of $V(G)$, we can find a valid soln of SMALL DISCONNECTED large subgraph in poly time

Proof: Consider any good partition (L, R). Now, consider $\vec{F_1}$, the component containing C_1 and $\vec{F_2}$, the component containing C_2. Clearly $\vec{F_1}$ and $\vec{F_2}$ have size at least k. Also, $\vec{F_1} \neq \vec{F_2}$, since in the original graph vertex C_1 and C_2 were not connected, hence vertices in $\vec{F_1}$ and $\vec{F_2}$ are also not connected.

Now, we can run min st vertex cut on each pair of vertices s, t, such that $s \in \vec{F_1}$ and $t \in \vec{F_2}$. Clearly since X itself is a valid such cut, we will only find a smaller vertex cut than X. Thus in $O(n^2)$ flow computations we can find such a valid soln.
This is polytime in n.

Intuition

If it is a yes instance, we could randomly assign vertices to L with prob. $1/2$ & vertices to R with $1/2$ prob.

$P(\exists$ a good partition$) \geq \frac{1}{2}^{2q+k}$

By repeating this $2^{2q+k} n^{O(1)}$ many times we would get constant success probability. Now we have to derandomize this.

Lemma 3: Consider $(n, k+2q)$-universal set $\mathcal{U}$. For partitions $L = T$ and $R = V(G) \setminus T$ for sets $T \in \mathcal{U}$, at least one of them is a good partition if it is a yes instance.

Proof: If it is yes instance, $\exists$ sets X, C_1, C_2 as defined in the good partition definition. Now, $|X \cup C_1 \cup C_2| = k+2q$.

Let $S = X \cup C_1 \cup C_2$. Since $|S| = k+2q$, we know all 2^{k+2q} subsets of S lie the family $\{A \cap S : A \in \mathcal{U}\}$. Thus, there $\exists\ T \in \mathcal{U}$ s.t. $X \subseteq T$ and $C_1 \cup C_2 \subseteq V(G) \setminus T$. This $L = T$ & $R = V(G) \setminus T$ is a good partition.

We can find the universal set $\mathcal{U}$ in lemma 3 in $2^{k+2q} (k+2q)^{O(\log k)} (\log n) n \leq 2^{O(k+q)} n \log n$ time. This gives us the desired result.

7. $\boxed{\text{10 marks}}$ **(Take Home Question.)** In the FEEDBACK VERTEX SET/TW problem we are given a graph G on n vertices and a tree decomposition $(T, \beta : V(T) \rightarrow 2^{V(G)})$. The goal is output the minimum size of a set $X \subseteq V(G)$, such that $G - X$ is a forest.

Design a $k^{O(k)} \cdot n^{O(1)}$-time algorithm for FEEDBACK VERTEX SET/TW. *You are required to define the table entries, computations at the base case(s), and using recursive formulae, prove the correctness, and do the runtime analysis. Also, show how using the computed entries, you can solve the problem at hand.*

Let $dp[t, X, P]$, $X \subseteq X_t$, P is a partition of $X_t \setminus X$ denote the state, such that

$dp[t, X, P] = \min |\bar{X}|$, s.t. $X \subseteq \bar{X} \subseteq V_t$, $\bar{X} \cap X_t = X$, for each connected component of $G_t \setminus \bar{X}$ $C_1, C_2, \dots C_i$. $C_i \cap X_t = P_i$

We always set $dp[t, X, P] = \infty$ if the components of $G[X_t \setminus X]$ have cycles. Add an arbitrary vertex say u^* to every bag

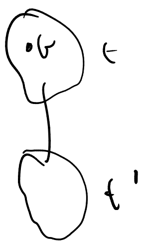
Base Case

Leaves :

$dp[t, u^*, \phi] = 0$, since the empty graph is acyclic.

$dp[t, \phi, \{\{u^*\}\}] = 0$, since it is a isolated vertex.

Recursive



Introduce Vertex Node : Say vertex 'v' is introduced.

$$dp[t, X, P] = \begin{cases} dp[t', X \setminus \{v\}, P] + 1 & \text{if } v \in X \\ dp[t', X, P \setminus \{v\}] & \text{if } v \notin X \text{ \& } \exists \text{ a partition containing only } v \\ \infty & \text{otherwise} \end{cases}$$

$P - v$ denotes the partition P', s.t. $P' = \{p \setminus \{v\} : p \in P\}$.

Correctness

* If $v \in X$,

Proving $dp[t, X, P] \leq dp[t', X \setminus \{v\}, P] + 1$

Let $\bar{X}$ be the opt. soln. for $dp[t', X - \{v\}, P]$. Clearly, $G_{t'} \setminus \bar{X}$ is acyclic. Note that since v is isolated vertex, $\boxed{\bar{X} \cup \{v\}}$ is also a valid FVS of G_t satisfying the requirements. Hence the optimum of the dp-state is at most $dp[t', X \setminus \{v\}, P] + 1$, since $\bar{X} \cup \{v\}$ has size $dp[t', X \setminus \{v\}, P]$

Proving $dp[t, X, P] \geq dp[t', X \setminus \{v\}, P] + 1$

Let $\hat{X}$ be the optimum soln. for $dp[t, X, P]$. Now, since $X \subseteq \hat{X}$, $v \in \hat{X}$. Since v is isolated vertex, $\hat{X} \setminus \{v\}$ is a FVS of G_t & hence $G_{t'}$. Thus, the bound follows.

* <u>If $v \notin X$</u> & $\exists$ a partition containing only v

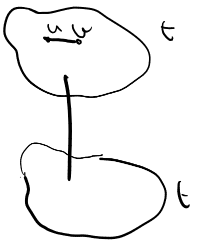
Note that $\{v\}$ is a connected component of its own. Thus, if $\hat{X}$ is a FVS for G_t, $\hat{X}$ is a FVS for $G_{t'}$ and vice-versa. So the dp holds.

* <u>If $v \notin X$</u> & $\nexists$ a partition containing only v

Clearly v forms a connected component of its own in G_t. Thus, if a partition contains some other vertices along with v, it is an "invalid" partition & $\nexists$ a set $\hat{X}$ satisfying the dp requirements.

<u>Introduce Edge Node</u>

Let $u-v$ be the edge introduced.



$$dp[t, X, P] = \begin{cases} dp[t', X, P] & u \notin X \text{ or } v \notin X \\ dp[t', X, P \setminus P_{uv} \cup \{P_u, P_v\}] & u \in X \;\&\; v \in X \;\& \\ & \text{if } u \;\&\; v \text{ are in same partition} \\ +\infty & \text{otherwise} \end{cases}$$

P_{uv} is partition containing

P_u & P_v are the partitions containing u & v respectively in $G[X_{t'}]$

Correctness

If v or u belong to X,

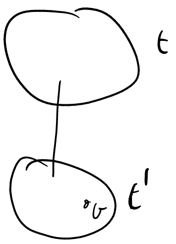
Anyways the edge uv will be deleted. Thus whatever partition of $V_t \setminus X$ we obtain for $G_{t'}$, the same are valid for G_t, and thus the DP holds.

$u \notin X$ & $v \notin X$ if u & v are NOT in same partition

Since we have an edge, they must be in the same component. Thus, the partition is invalid.

$u \notin X$ & $v \notin X$ if u & v are in same partition

Clearly, since the edge u-v is newly added, in G_t, u & v must belong to separate components to ensure it is acyclic. Thus in $G_{t'}$, we look for the partition where u & v are in separate components, and thus the dp values have to be equal.



Forget Node

$$dp[t, X, P] = \min\left(dp[t', X \cup \{v\}, P], \min_{P' \in \mathcal{P}} \left(dp[t', X, P' \right)\right)$$

$$\mathcal{P} \leftarrow \{ P' : \exists\, p \in P' \text{ s.t. } P = (P' \setminus p) \cup (p \setminus \{v\}),\ p \mid \{v\} \in P \text{ or } p = \{v\} \}$$

-> Set of all partitions which are "similar" to P, except that v belongs to one of the partitions or v belongs to a partition of its own.

Correctness:

$$dp[t, X, P] \leq \min\left(dp[t', X \cup \{v\}, P], \min_{P' \in \mathcal{P}} \left(dp[t', X, P' \right)\right)$$

The optimum soln $\vec{x}$ for $dp[t', X \cup \{v\}, P]$ as well as $dp[t', X, P']$ $\forall P' \in P$ is also a valid FVS for $dp[t, X, P]$, thus they form an upper bound on $dp[t', X, P]$.

$$dp[t, X, P] \geq \min\left(dp[t', X \cup \{v\}, P], \min_{P' \in P}\left(dp[t', X, P'] \right)\right)$$

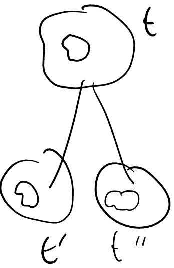
Let $\vec{x}$ be the optimum for $dp[t, X, P]$. Now, either $v \in X$ or it doesn't. If $v \notin X$, $\exists$ a partition $P' \in P$ such that v belongs to one of the partitions in P'.

Thus, $\vec{x}$ must be optimal for either the case $v \in X$, i.e. $dp[t', X \cup \{v\}, P]$, or one of the partitions $P' \in P$ $dp[t', X, P]$.

Thus,

$$dp[t, X, P] \geq \min\left(dp[t', X \cup \{v\}, P], \min_{P' \in P}\left(dp[t', X, P'] \right)\right)$$

holds true.



Join Node

$$dp[t, X, P] = dp[t', X, P] + dp[t'', X, P] - |X|$$

$$dp[t, X, P] \leq dp[t', X, P] + dp[t'', X, P] - |X|$$

Let $\vec{X_1}$ be the opt set for $dp[t', X, P]$ and $\vec{X_2}$ for $dp[t'', X, P]$.

Now, $\vec{X_1} \cup \vec{X_2}$ is a valid FVS for G_t. Reason: There are no edges across $G[V_{t_1} \setminus X_t]$ and $G[V_{t_2} \setminus X_2]$. Thus in the graph G_t, $\vec{X_1} \cup \vec{X_2}$ will be a valid FVS, since the cycles run only within G_{t_1} & G_{t_2}

Now since $|\hat{X_1} \cup \hat{X_2}| = dp[t', X, P] + dp[t'', X, P] - |X|$, since only $X = \hat{X_1} \cup \hat{X_2}$. Thus, $\hat{X_1} \cup \hat{X_2}$ is a solution for $dp[t, X, P]$ and is an upperbound.

$$\underline{dp[t, X, P] \geq dp[t', X, P] + dp[t'', X, P] - |X|}$$

Let $\hat{X}$ be opt. set for $dp[t, X, P]$. Clearly, $\hat{X} \cap X_{t'}$ is a FVS for $dp[t', X, P]$. Thus, $|\hat{X} \cap X_{t'}| \geq dp[t', X, P]$

Likewise $|\hat{X} \cap X_{t''}| \geq dp[t'', X, P]$

$$|\hat{X} \cap X_{t'}| + |\hat{X} \cap X_{t''}| = dp[t, X, P] + |X| \geq dp[t', X, P] + dp[t'', X, P]$$

__Runtime__: There are ${}^{n}C_{n-i}$ ways to pick the set X, and i^i to partition $X_t \setminus X$ for a node t. Thus, total of $\sum_{i=0}^{n} {}^{n}C_i\, i^i \leq k^{O(k)}$

There are a total of $k^{O(k)}$ table entries at each node, thus $k^{O(k)} n^{O(1)}$ entries overall. It is trivial to see that each dp recursion can be $O(n)$ time. Thus, the algo runs in $k^{O(k)} n^{O(1)}$ - time.

__Solving problem__: For root r, output $\boxed{\min\left(dp[r, \{u^*\}, \phi],\ dp[r, \phi, \{u^*\}]\right)}$