1 Optimal Branching Systems (OBS)

Branching: A set of directed trees that don't share any edges (edge-disjoint), and collectively span the graph.
Branching system: (Not defined in lecture, but inferred by Nike). A set of branchings from r root nodes.
Optimal branching system (OBS): Given a directed graph with a cost for each edge and r specific root nodes, find (if they exist) a least expensive branching system from those nodes.
Convenient reduction of OBS: Find (if they exist) a least expensive set of k branchings which don't share any edges (edge-disjoint) all rooted at node r. There is a "beautiful, direct, easy algorithm" when $k = 1$,but we need matroid partitioning and intersections when $k > 1$.
Disjoint Branchings Theorem: The maximum size of a set B of edge-disjoint r-rooted spanning directed trees in G **equals** the minimum size of a set C of edges of G which are directed into a set S of nodes such that S is non-empty and does not contain root node r.

$$\max |B| = \min |C|, C = \{C \in G(E) : C \text{ are directed into a set } S \in V(G), S \neq \emptyset, r \notin S\ \} \tag{1}$$

Good algorithms for finding B and C were devised by Lovasz and by Tarjan.
Corollary: If a set $D \in G(E)$ such that:

- (a) D has no edges entering r, and exactly k edges entering each other node of G. (D is?) a basis of a matroid M_1;
- (b) D can be partitioned into k edge-disjoint spanning trees of G. (D is?) a basis of M_2;

then, D can be partitioned into k edge-disjoint spanning trees of G, each rooted at r.
Analogy: When the capacity of each edge is 1, a min-cost flow algorithm finds a set of edges which still needs to be partitioned into edge-disjoint paths from source to sink, but it's more complicated for spanning directed trees rooted at node r.

So finding an OBS entails finding a min-cost set D of edges which is both a basis of matroid M_1 and of M_2 (optimum matroid intersection problem).

- $M_1 = (E, J)$ in which J has no edges entering node r, and at most (at most? but corollary said "exactly") k edges entering each other node of G. Edmonds says it's "obvious" that this is a matroid.
- $M_2 = (E, J)$ in which J can be partitioned into at most k forests. This being a matroid is a corollary of **matroid partitioning**.

Algorithm:

- Find forests in the graph (linearly independent columns of a matrix in GF(2)?)
- Use a matroid partitioning algorithm to recognize sets of columns which are partitionable into k independent sets.
- Matroid partitioning is used as a subroutine to identify independent sets (the J) of M_2.
- Then do optimum matroid intersection on M_1 and M_2.

Directed Tree: Eric: You sure the arrows don't all have to go in same direction?
Spanning Subgraph: Eric: are you sure about this definition? It literally doesn't matter at all which edges you include?