Clustering is an approach for grouping similar items, for example grouping co-regulated genes that contribute to the same process in a cell. In general we usually have experimentally obtained a set of pairwise similarities of items, and we are interested in a hidden clustering closely representing this data.

Let us consider the following abstraction of this problem. Given a graph $G = (V, E)$ where the vertices represent the items and each edge represents a pairwise similarity between the items. We assume we are given a nonnegative integer k. The question is then to find out whether we can transform G, by deleting or adding at most k edges, into a graph that consists of a disjoint union of cliques. (Recall that a clique is a subgraph induced by a set of vertices that are all mutually connected by the edges.) See Figure 1 for an example where any $k \geq 2$ leads to the answer yes (because one edge –in the center– needs to be deleted and one edge –at the bottom-left– needs to be added).

We are interested in finding a bounded search tree to solve this problem. The idea is to use the following lemma.

Lemma. A graph $G = (V, E)$ is a disjoint union of cliques *if and only if* there are no three distinct vertices u, v, $w \in V$ with $\{u, v\} \in E$ and $\{u, w\} \in E$, but $\{v, w\} \notin E$.

1. (4 points) Give a proof of this lemma (*both* directions).

Solution: From left to right:

1. Suppose that G is a collection of disjoint cliques.
2. To arrive at a contradiction, suppose it has three distinct vertices $u, v, w \in V$ with $\{u, v\} \in E$ and $\{u, w\} \in E$, but $\{v, w\} \notin E$.
3. Then the sub-graph induced by $\{u, v, w\}$ is not a clique, but the nodes are not disjoint either.
4. Therefore there must be no such three vertices in V.

An alternative proof from left to right, directly from the definition:

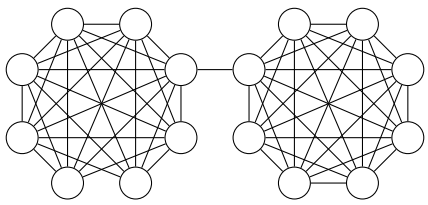

Figure 1: An example of a graph that can be edited into a disjoint union of cliques by removing just one edge (the one in the middle) and adding one other (bottom of left cluster).

1. Suppose that G is a collection of disjoint cliques.
2. Let a vertex $u \in V$ be given with at least two neighbors $v, w \in V$. If there is no such vertex v, the lemma is vacuously true.
3. Now, u, v, w must be all in the same clique, because the graph is a disjoint union of cliques and these vertices are connected.
4. Now, $\{v, w\}$ must be an edge $\in E$ from the definition of a clique.
5. So there cannot be a triple of vertices with $\{v, w\} \notin E$.

From right to left using contraposition:

1. Suppose that a graph G is not a collection of disjoint cliques.
2. Then there exists a connected subgraph of G with at least 3 vertices that is not a clique. (Because subgraphs with 1 or 2 vertices are cliques by definition.)
3. In this subgraph, there are vertices $z, w \in V$ for which $\{z, w\} \notin E$.
4. Since the subgraph is connected, there exists a shortest path from z to w.
5. Consider this shortest path $z = u_0, u_1, ..., u_n = w$. The length of this path is at least two since $\{z, w\} \notin E$.
6. Now take $v = u_{n-2}, u = u_{n-1}, w = u_n$. Then we have three distinct vertices $u, v, w \in V$ with $\{u, v\} \in E$ and $\{u, w\} \in E$. It also holds that $\{v, w\} \notin E$ since we use the shortest path.

An alternative, more direct proof, from right to left:

1. Suppose there are no three distinct vertices u, v, $w \in V$ with $\{u, v\} \in E$ and $\{u, w\} \in E$, but $\{v, w\} \notin E$.
2. Let C be any of the connected components.
3. Consider vertices $x, y \in C$.
4. Let $x = u_0, u_1, ..., u_n = y$ be a path between them.
5. We show that x and y are neighbors by shortening this path iteratively:
   (a) Choose an $0 < i < n$ (if it exists)
   (b) Consider $v = u_{i-1}$, $u = u_i$, $w = u_{i+1}$: it then must be that $\{v, w\} \in E$

(c) u_i can thus be removed from the path; reindex vertices and repeat until no such $0 < i < n$ exists

6. Thus $x = u_0$ and $y = u_n$ must be neighbors.
7. Since this holds for any x and y, all vertices in C are neighbors, so C is a clique.
8. This holds for every connected component C, so all are cliques.

2. (4 points) Use this lemma to define a search-tree-based algorithm for a given integer k. (Describe all sub-cases by recursive calls and do not forget to describe the base case(s).)

Solution: The following bounded search tree algorithm finds out whether a graph G can be transformed in a disjoint union of cliques by adding or deleting at most k edges.

1. If the graph G is already a union of cliques then we are done. Return "yes".
2. If $k \leq 0$ then return "no".
3. Otherwise, identify vertices u, v, $w \in V$ with $\{u, v\} \in E$ and $\{u, w\} \in E$, but $\{v, w\} \notin E$ (which we now know exist), and recursively solve the following three sub-problems on graphs $G' = (V, E')$ with the nonnegative integer $k' = k - 1$.
   (a) $E' := E - \{\{u, v\}\}$
   (b) $E' := E - \{\{u, w\}\}$
   (c) $E' := E \cup \{\{v, w\}\}$

   Then return "yes" if at least one of these returns "yes"; return "no" otherwise.

3. (2 points) Derive a recursive formula (recurrence relation) for an upper bound on the run time depending on k (so for $T(k)$) and then derive an upper bound on this run time using $O^*()$ notation (in closed form).

Solution: The recursive algorithm described above has a run-time of $T(k) \leq 3 \cdot T(k-1) + O(m+n)$. This leads to a search tree of depth k with a branching factor of 3, so the run-time is $O^*\left(3^k\right)$.

(The $O(m+n)$ comes from an algorithm that verifies for each cluster whether it is a cluster, e.g., this involves verifying whether each neighbor w of a vertex v has the same set of neighbors (except for the vertex itself), i.e., $N(w) \cup \{w\} = N(v) \cup \{v\}$ for all $w \in N(v) \cup \{v\}$.)

End of assignment