Sources

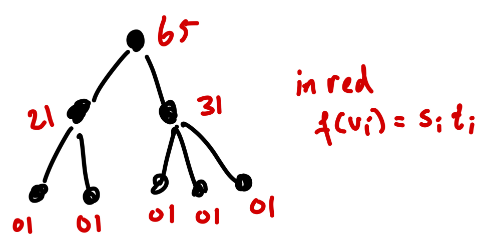
Diestel's Graph theory: https://diestel-graph-theory.com

Dynamic Programming

Dynamic programming is an optimization method, where a problem is phrased as a set of functions $f(1), \dots, f(n)$ where each function $f(i)$ can be calculated by a simple optimization based on $f(1), \dots, f(i-1)$.

We consider the following example of determining $\alpha(T)$ for some tree T. This problem is NP-hard on general graphs, as we saw last week. On class, we discussed a few polynomial time algorithms for finding maximum independent sets on trees (for example, by König's Theorem 2.1.1 in Diestel, or by exploiting the fact that maximum independent sets may contain all leaves). Here, we will use a dynamic programming approach. Consider the tree to be rooted at a vertex v_n. Let $f(v_i) = (s_i, t_i)$ so that s_i is the size of a largest independent set in the subtree rooted at v_i over all such sets that do not contain v_i, and t_i the size of a largest independent set over all such sets that do contain v_i. Now, note that the size of a largest independent set in T either contains v_n or does not contain v_n, and therefore $\alpha(T) = \max(s_n, t_n)$, with

$$s_n = 1 + \sum_{i, v_i \in N(v_n)} s_i, \qquad t_n = \sum_{i, s_i \in N(v_n)} \max(s_n, t_n).$$



Exercise 1. *Show that every tree T with root v has an ordering of the vertices $v_1, \dots, v_n = v$ such that if v_i is a child of v_j in the tree, $i < j$.*

Using the ordering of the vertices in the exercise, we now see that the independent set problem can be solved by dynamic programming. Take a moment to note why this exact approach fails in general graphs. There are ways to use dynamic programming for a large class of problems that include independent set, on more general classes of graphs, as long as they are "close" to being trees.

Tree decompositions

A *tree decomposition* tree T of a graph G is a tree, such that the vertices of T are sets $V_i \subseteq V(G)$, which are sometimes called "bags", and edges are such that T has the following properties:

(i) T is a tree,

(ii) for every $vw \in E(G)$, $v, w \in V_i$ for some $V_i \in V(T)$,

(iii) for every $v \in V(G)$, the subgraph of T induced by $\{V_i \mid v \in V_i\}$ is connected.

Note that every graph G has a trival tree decomposition obtained by simply taking all of its vertices in one bag. However, for our purposes we want to minimize the maximum size of a bag. This is referred to as the *treewidth* of a graph G, denoted $\mathrm{tw}(G)$, which is $k-1$, where k is the mimimim size of the maximum bag taken over all tree decompositions of G.

Claim 1. *For every complete graph K_n, we have $\mathrm{tw}(K_n) = n-1$. For any graph G not isomorphic to the complete graph, we have $\mathrm{tw}(G) < n-1$.*

Exercise 2. *Prove Claim 1. (We discussed an outline briefly in class.)*

Exercise 3. *Prove the following: consider a tree decomposition T of some graph G. For any three vertices $V_i, V_j, V_k \in V(T)$, such that V_j lies on the path between V_i and V_k, we have that $V_i \cap V_k \subseteq V_j$.*

Claim 2. *For any G, there exists an optimal tree decomposition T of G such that there are no two bags $V_j \neq V_k$ in $V(T)$ such that $V_j \subseteq V_k$.*

Proof. Suppose that a tree decompisition of G has such a pair V_j, V_k. Consider the path from V_j to V_k in G, and let V_h be the vertex on that path adjacent to V_j. Then, by Exercise 3, $V_j \subseteq V_h$. Now, we contract the edge V_jV_h to obtain another tree decomposition of G with the same treewidth. Repeating this process must terminate since the number of vertices of the tree decreases each time. □

Lemma 3. *A graph G of treewidth k has a vertex of degree at most k.*

Proof. We need the following fact about trees: every tree (or forest) has a vertex of degree at most 1. Let T be an optimal tree decomposition of G, then T has a vertex V_i of degree 1. If we assume T is as in Claim 2, then V_i contains a vertex v which does not appear in its neighbor in T, and therefore does not appear at all elsewhere in T. Then all of the edges incident to v must be represented in V_i, and since $|V_i| \leq \mathrm{tw}(G) + 1$, we have $d(v) \leq k$. □

We can make a stronger statement here. In general, if we delete a vertex v from a graph G, we can obtain a tree decomposition T_{G-v} by deleting v from any bag it appears in in T_G. Therefore, we see that $\mathrm{tw}(G-v) \leq \mathrm{tw}(G)$. This means that we can repeatedly delete vertices of degree $\leq k$ as in Lemma 3.

Claim 4. *For any n-vertex G with $\mathrm{tw}(G) = k$, there exists an ordering on its vertices $v_1, \ldots, v_n$ such that $|N(V_i) \cap \{v_1, \ldots, v_{i-1}\}| \leq k$.*

The following Corollary is along the lines of Brooks' Theorem in 5.2 of Diestel.

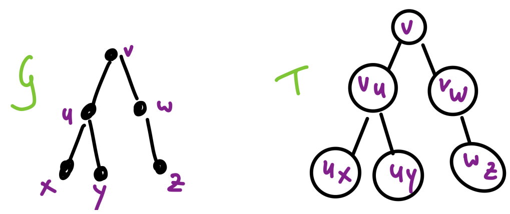
Corollary 5. *For any graph G, we have*

$$\chi(G) \leq \mathrm{tw}(G) + 1.$$

Proof. Use the ordering in Claim 4 and greedily assign each vertex a color from the set $\{1, 2, \ldots, \mathrm{tw}(G) + 1\}$, avoiding colors previously assigned to its neighbors. Each vertex has at most k neighbors previously assigned and must therefore have an available color in the set. □

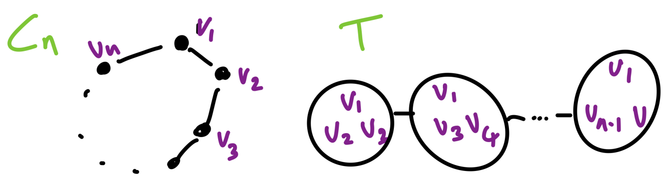
Lemma 6. *A graph G has treewidth 1 if and only if it is a forest.*

Proof. We work by induction on $n = |G|$. Clearly, this holds when $n = 1$. If G has treewidth 1, then by Lemma 3, G has a vertex v of degree 1. It is easy to see that G is a forest if and only if $G - v$ is a forest. We obtain a tree decomposition of treewidth at most 1 of $G - v$ by removing v from its bags in T, and then we are done by the I.H.
If G is a tree, we obtain a tree decomposition as follows. Let G be rooted at a vertex v. We build a tree T isomorphic to G via the following bijection: $f(v) = \{v\}$, and for all other vertices $w \in V(G) \setminus v$, $f(w) = \{w, p(w)\}$, where $p(w)$ is the parent of w in G rooted at v. Clearly, every edge is represented in this decomposition, and every vertex induces a star on its bags. □



Lemma 7. *For any cycle C_n, we have* $\text{tw}(C_n) = 3$.

Proof. The lower bound is given by Lemma 6, and we give the upper bound by the following construction:



□

Series-parallel graphs

Cycles are not the only graphs of treewidth 2. This class is a larger class of graphs known as series-parallel graphs. Often these are defined recursively as follows. K_2 is series-parallel, with one vertex v_s designated as the source and one v_t as the sink. One obtains a new series-parallel graph G from two series-parallel graphs G_1, G_2 as follows. Let v_s, v_t be the source and sink of G_1, respectively, and let w_s, w_t be the source and sink of G_2. Now join the graphs in one of two ways:

- In series: join G_1 and G_2 by identifying v_t and w_s, and let v_s, w_t be the source and sink of the new graph G.
- In parallel: join G_1 and G_2 by identifying v_s and w_s as the source of the new graph G, and identifying v_t and w_t as the sink of the new graph G.

This intuitive definition gives the class needed in many applications. However in this setting, our family has to be a bit larger, and needs to contain all subgraphs of graphs obtained as above. Instead we use the following definition.

Definition 8. *Series-parallel are graphs that can be obtained from K_1 by any number of the following operations:*

- *Add a an isolated vertex,*
- *add a vertex with one edge to an existing vertex,*
- *add a self-loop,*
- *add an edge parallel to an existing edge,*
- *subdivide an edge (replace it with a path of length 2 via a new vertex).*

Exercise 4. *Show that the class of series-parallel graphs is closed under taking induced subgraphs. (This is equivalent to showing that this class is closed under vertex-deletion.)*

Now, we will show that the class of series-parallel graphs is exactly the class of graphs with treewidth at most 2.

Theorem 9. *G is a series parallel graph* $\Leftrightarrow \mathrm{tw}(G) \leq 2$.

Proof. Let G be an n-vertex series-parallel graph and consider a construction as in Definition 8. We will build a tree decomposition along with it. We start from K_1 with a single bag containing the first vertex. Note that adding a self-loop or parallel edge does not require any changes to a given tree-decomposition. Let v be the last vertex added. If v is an isolated vertex, we add a bag $\{v\}$ with an edge to any exsisting bag. If v is added with one edge to an existing vertex w, we add a bag $\{v, w\}$ with an edge to any bag containing w in the current tree. If v is added as the internal vertex of a subdivided edge xy, we note that x and y must appear together in some bag V_i. We add a new bag $\{x, y, v\}$ with an edge to V_i. In each case, verify that the decomposition remains valid.

We will work by induction on $|G|$ to prove the reverse direction. The base case, K_1 is straightforward. Now, suppose that G is a graph with a tree decomposition T of width 2. Since T is a tree, it has a leaf vertex V_i with $|V_i| \leq 3$. We will assume as before that no bag contains another. If $|V_i| = 1$, then its vertex v must be an isolated vertex in G, and G is series parallel if and only if $G - v$ is series parallel. Observe that $T - V_i$ is a tree decomposition of $G - v$ of width at most 2, so the result holds by the inductive hypothesis. Similarly, if $|V_i| = 2$, as before, one of its two vertices v, w does not appear elsewhere and is leaf. Again, G is series parallel if and only if $G - v$ is series-parallel, and $T - V_i$ is a tree decomposition of $G - v$ of width at most 2. Finally, if $|V_i| = 3$, let w be a vertex in V_i that does not appear elsewhere. Consider the tree decomposition obtained from deleting w from V_i. This is a valid tree decomposition for the graph $G - w$ with or without the edge xy. Therefore, by the inductive hypothesis $(G-w)+xy$ is a series-parallel graph, and G is obtained by subdividing the edge xy (after copying it if necessary).

□

Planar graphs and grids

We define the $n \times m$ *grid graph* $P_n \square P_m$ as follows, This graph has vertices v_{ij} for $1 \leq i \leq n$ and $1 \leq j \leq m$, such that $v_{ij}v_{kl}$ for distinct vertices is an edge if either $i = k$ or $j = l$.

Exercise 5. *Give the possible values of* n, m *such that* $P_n \square P_m$ *is series-parallel.*