Homework 2

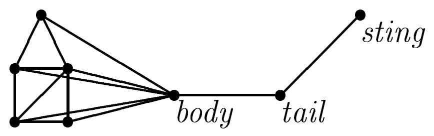
1. Give a linear-time algorithm for topological sort based on depth-first search.

2. Let $G = (V, E)$ be an undirected graph, and let θ be a circular ordering of the edges adjacent to each vertex. The ordering θ is said to be *consistent* with an embedding of G in the plane if for each vertex v, the ordering of the edges adjacent to v given by θ agrees with their counterclockwise ordering around v in the embedding.

   (a) Give a linear-time algorithm that determines whether a given θ is consistent with some plane embedding.

   (b) Consider two embeddings to be the same if one can be transformed into the other by a smooth motion in the plane without tearing or cutting. Assume that G is connected. How many distinct plane embeddings are consistent with a given θ? (*Extra credit.* Remove the assumption of connectedness.)

3. A *scorpion* is an undirected graph G of the following form: there are three special vertices, called the *sting*, the *tail*, and the *body*, of degree 1, 2, and $n - 2$, respectively. The sting is connected only to the tail; the tail is connected only to the sting and the body; and the body is connected to all vertices except the sting. The other vertices of G may be connected to each other arbitrarily.



Give an algorithm that makes only $O(n)$ probes of the adjacency matrix of G and determines whether G is a scorpion. (This is a counterexample to an earlier version of the famous *Anderaa-Rosenberg conjecture*, which stated that any nontrivial graph property that is invariant under graph isomorphism requires $\Omega(n^2)$ probes of the adjacency matrix to test. Anderaa disproved this version in 1975 using a different counterexample (see [91]), but conjectured that it held for *monotone* properties (those that cannot change from false to true when edges are deleted). This was later verified by Rivest and Vuillemin [91].)

We claim that with c components this number is

$$(1 + \sum_{i=0}^{c-1} (f_i - 1))^{c-1} \ .$$

Each embedding determines a labeled forest whose vertices are the components. A component i is a child of another component j in this forest if it occurs inside an inner face of j with no intervening components. The component i can be placed inside any one of $f_j - 1$ inner faces of j; let us indicate this by labeling the edge (i, j) in the forest with the factor $f_j - 1$. Thus the number we are seeking is the sum over all labeled forests of the product of the edge labels on that forest.

We modify an argument of Prüfer that there are n^{n-2} labeled trees on n nodes (see [62]). We establish a one-to-one correspondence between labeled forests and sequences of length $c - 1$ from the set

$$\{\top, 0, 1, \ldots, c-1\} \tag{2}$$

as follows. Given a labeled forest, start with the null sequence and repeat the following operation until there is only one vertex left. Prune off the lowest numbered leaf i and append j to the sequence, where j is the parent of i. If i has no parent, then append $\top$ to the sequence. Then each labeled forest determines a sequence of length $c-1$, and it is not difficult to reconstruct the forest uniquely from the sequence.

For each sequence $a_0, a_1, \ldots, a_{c-2}$ of length $c - 1$ over the set (2), the product of the edge labels on the labeled forest corresponding to that sequence is

$$g(a_0)g(a_1) \cdots g(a_{c-2}) \ ,$$

where $g(i) = f_i - 1$ and $g(\top) = 1$. The number we are seeking is the sum of all such products. This number is

$$(g(\top) + g(0) + g(1) + \cdots + g(c-1))^{c-1} \quad = \quad (1 + \sum_{i=0}^{c-1} (f_i - 1))^{c-1} \ .$$

The total number of embeddings is this number times the number of ways of choosing the outer faces of the components, or

$$(1 + \sum_{i=0}^{c-1} (f_i - 1))^{c-1} \prod_{i=0}^{c-1} f_i \ .$$

3. Here is an algorithm that takes $5n$ probes of the adjacency matrix. Let $d(v)$ be the degree of vertex v. The main difficulty is to locate one of the

interesting vertices (the body, tail, or sting); once we have done that, we can locate all the other interesting vertices with $3n$ probes and check that the graph is a scorpion. For example, if we have found a vertex v with $d(v) = n - 2$, then that vertex must be the body if the graph is a scorpion. By scanning the v^{th} row of the matrix, we can check that $d(v) = n - 2$ and determine its unique non-neighbor u, which must be the sting if the graph is a scorpion. Then by scanning the u^{th} row, we can verify that $d(u) = 1$ and find its unique neighbor w, which must be the tail; and with n more probes we can verify that $d(w) = 2$.

We start with an arbitrary vertex v, and scan the v^{th} row. If $d(v) = 0$ or $n - 1$, the graph is not a scorpion. If $d(v) = 1$, 2, or $n - 2$, then either v is interesting itself or one of its 1 or 2 neighbors is, and we can determine all the interesting vertices as above and check whether the graph is a scorpion with at most $4n$ additional probes.

Otherwise, $3 \leq d(v) \leq n-3$, and v is boring. Let B be the set of neighbors of v and let $S = V - (B \cup \{v\})$. The body must be in B and the sting and tail must be in S. Choose arbitrary $x \in B$ and $y \in S$ and repeat the following: if x and y are connected, then delete y from S (y cannot be the sting) and choose a new $y \in S$. If x and y are not connected, then delete x from B (x is not the body unless y is the sting) and choose a new $x \in B$. If the graph is indeed a scorpion, then when this process ends, B will be empty and y will be the sting. To see this, observe that B cannot be emptied without encountering the sting, because the body cannot be deleted from B by any vertex in S except the sting; and once the sting is encountered, all remaining elements of B will be deleted.

Whether or not the graph is a scorpion, the loop terminates after at most n probes of the adjacency matrix, since after each probe some vertex is discarded.

If a property is such that we have to look at *every* entry in the adjacency matrix, then that property is said to be *evasive*. Many monotone graph properties have been conjectured to be evasive. Yao [110] has shown that all monotone bipartite graph properties are evasive if we are given a bipartite adjacency matrix representation. The question for general graphs remains open. Bollobás discusses this issue in his book [13]. He gives a $6n$-probe solution to the scorpion problem there.