Homework 1
Due: Wednesday, November 3 in LaTeX

1: Orientability

A basic operation on a representation of a graph is the *adjacency query:* given two vertices u and v, tell whether the edge (u, v) is present in the graph.

Two common representations of a graph $G = (V, E)$ are *adjacency matrices* and *adjacency lists*. The adjacency matrix supports adjacency queries in time $O(1)$, but requires space $O(|V|^2)$. Adjacency lists only require space $O(|E| + |V|)$, but adjacency queries may take time $O(|V|)$. In the following we study how to obtain $O(1)$ time adjacency queries in space $O(|E| + |V|)$ for a large class of sparse graphs.

Let $G = (V, E)$ be an undirected graph. If the edges in E can be assigned orientations such that each vertex in V has out-degree at most c, the graph G is said to be *c-orientable.*

a) Prove that G is c-orientable if and only if

$$\max_J \frac{|E(J)|}{|V(J)|} \leq c ,$$

where J ranges over all subgraphs of G. Here, $|V(J)|$ is the number of vertices in J and $|E(J)|$ is the number of edges.

b) Give an algorithm that from an adjacency list representation of a c-orientable graph $G = (V, E)$ computes a c-orientation of G in time $O(|V| + |E|^2)$.

c) Give an algorithm that from an adjacency list representation of a c-orientable graph $G = (V, E)$ computes a $2c$-orientation of G in time $O(|E| + |V|)$.

d) Show how to represent a c-orientable graph $G = (V, E)$ using space $O(|E| + |V|)$ and preprocessing time $O(|E|+|V|)$ such that adjacency queries can be answered in time $O(c)$.

e) Prove that any planar graph is 3-orientable (Hint: Euler's formula).

2: Array initialization

Describe a data structure supporting the following operations in worst-case constant time:

- $\mathsf{Init}(n)$ – create an array A of length n with all entries set to nil.
- $\mathsf{Set}(i, v)$ – set the ith entry of A to v.
- $\mathsf{Get}(i)$ – return the ith entry of A.

The operations may use a procedure $\mathsf{alloc}(n)$ that in worst-case constant time returns an uninitialized array of length n, *i.e.*, all entries of the allocated array contain random information.

3: Sink

Given the adjacency-matrix of a directed graph $G = (V, E)$, show how to compute in $O(|V|)$ time if G has a *sink*, *i.e.*, a vertex of in-degree $|V| - 1$ and out-degree 0.

4: Scorpion *

A *scorpion* is an undirected graph $G = (V, E)$ of the following kind: there are three special vertices called the *sting*, the *tail* and the *body* of degree 1, 2 and $|V| - 2$, 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. Besides being connected to the body, the remaining $|V| - 3$ vertices may be connected to each other arbitrarily.

Give an algorithm that makes only $O(|V|)$ probes of the adjacency matrix of G and determines whether G is a scorpion.