1. ***Introduce yourself.*** Why do you take the course? What is your personal goal and expectations? Which aspects of planar graphs might potentially be useful to you? How much time and energy do you plan to invest this term?

2. ***Graph Laplacians.*** Let G be an undirected graph and A be its adjacency matrix. Define the degree matrix D to be the diagonal matrix where $D[i,i] = \deg(v_i)$ for every vertex v_i and $D[i,j] = 0$ for every $i \neq j$. The ***Laplacian*** of graph G is defined to be

$$L := D - A.$$

   (a) Prove that the quadratic form $x^T L x$ is non-negative for any real vector x.

   (b) Prove that for any graph G, the smallest eigenvalue λ_1 of L is always 0, using the variational characterization of eigenvalues:

$$\lambda_1 = \min_{x \neq 0} \frac{x^T L x}{x^T x}.$$

   (c) Prove that for any graph G, the second smallest eigenvalue λ_2 is 0 if and only if G is disconnected.

Vectors and matrices will appear a few times in this class, so it would be helpful to equip yourself with basic knowledge about them. There is no need to go through a whole textbook on linear algebra; sufficient knowledge on the following items will be good enough:

- *eigenvalues and eigenvectors, quadratic forms, Courant–Fischer variational definition*
- *linear programming (only in one topic)*

3. ***Winter is coming.*** A *bush* T is an unordered tree; we refer to edges as *branches*, and degree-1 nodes as *leafs*. We assume that every node of T has degree at most 3.

   (a) Prove that there is a branch e^* in an n-node bush T such that each tree component of $T - e^*$ has at least $n/4$ nodes.

   Such a branch e^* is called *balanced*. We can *cut* a bush T into two by choosing a branch e^* and cut T into $T - e^*$. Consider the following algorithm FIREWOOD.

```
FIREWOOD(T, r):
  if T has at most r nodes:
      return singleton set {T}
  find a balanced branch e* in T
  cut the bush T at e* into T_1 and T_2
  return FIREWOOD(T_1, r) ∪ FIREWOOD(T_2, r)
```

   Figure 1. Pseudocode for Firewood, cutting the original bush into roughly equal-sized chunks.

   A node is *trimmed* if at least one of its incident branches was ever cut during the execution of FIREWOOD.

   (b) Prove that the number of trimmed nodes in the collection of bushes returned by FIREWOOD is at most $O(n/r)$.

   (c) Prove that the number of trimmed nodes on a single bush returned by FIREWOOD can be as large as $\Theta(\log n)$. Why is this not a contradiction to (b)?

   *(d) Design an different algorithm to chop the bush T so that every bush returned has at most r nodes and $O(1)$ many trimmed nodes, where the constant is independent to r.

Basics in algorithm analysis will be very useful for this course. You need the following:

- *reading pseudocode, correctness/time analysis using asymptotics*
- *induction, binary search, breadth-first search, shortest-path algorithms*

We will also encounter elementary probability stuff like events and expectation, nothing fancy.