# Problem Set 2

This problem set is due Wednesday, September 28 at noon.

1. Prove that any undirected planar graph $G$ with non-negative edge weights can be transformed into an undirected planar graph $G'$ with maximum degree 3 such that,

   - for any $u, v \in V(G)$, $d_G(u, v) = d_{G'}(f(u), f(v))$, where $f : V(G) \to V(G')$ maps vertices between $G$ and $G'$; and
   - $|V(G')| = O(|V(G)|)$.

   **Solution:** We replace each node $u$ of degree $d$ by a cycle $C_u$ on $d$ nodes. Each edge $uv$ is represented by *(1)* one node $c_{uv}$ on the cycle $C_u$, *(2)* one node $c_{vu}$ on the cycle $C_v$, and *(3)* an edge $c_{uv}c_{vu}$. The edge $c_{uv}c_{vu}$ is assigned the weight of $uv$. The edges on the cycle have zero weight. $f$ maps any node $u$ to an arbitrary node of $C_u$. Shortest-path distances are maintained. For each edge we introduce two nodes. The number of nodes in $G'$ is thus $O(|E(G)|) = O(|V(G)|)$.

2. A $\rho$–*clustering* of $G$ is a decomposition into $O(n/\rho)$ vertex-disjoint *connected pieces*, each with $\Theta(\rho)$ vertices. Recall that a $\rho$–clustering, if computed efficiently, can be used to compute an $r$–division in $o(n \log n)$ time. Give a linear-time algorithm to compute a $\rho$–clustering for any connected graph with maximum degree three.

   **Solution:** This solution is also described in a paper by Greg N. Frederickson entitled "Data structures for on-line updating of minimum spanning trees" (SICOMP 1985).

   We begin with computing any spanning tree $T$ of $G$. Note that $T$ has maximum degree three. Starting from an arbitrary leaf of $T$, we traverse $T$ in *depth-first order*. The following recursive procedure CSearch, called for a vertex $v$, generates clusters of size between $\rho$ and $2\rho - 1$ and (potentially) returns one "remainder" set of size $< \rho$.

   CSearch($v$)

   - $C := \{v\}$
   - For Each child $w$ of $v$ Do: $C := C \cup$ CSearch($w$)
   - If $|C| < \rho$: Return $C$
   - Else: Output $C$ and Return $\emptyset$

   In the end, we create the union of the last cluster output with the set returned by CSearch to ensure that all the clusters have size at least $\rho$.

   Clusters form connected components (they are connected in $T$). Since we started at a leaf and since $T$ has degree at most three, each vertex has at most two children. CSearch returns sets of size $\leq \rho - 1$. At any node $u$ with children $v$ and $w$, the union of $\{u\}$, CSearch($v$), and CSearch($w$) has size at most $2\rho - 1$. The final cluster may consist of the union of a set of size $\leq \rho - 1$ and a cluster of size $\leq 2\rho - 1$, which has total cardinality $3\rho - 2$. As a consequence, clusters have sizes between $\rho$ and $3\rho - 2$.