Exercise 2.1:
In this exercise, we will prove that the following problem is NP-complete: Let $G = (V, E)$ be an undirected graph. Is there a function

$\alpha : E \rightarrow \{1, 2, 3\}$ s.t. for any vertex $v \in V$ and any $e, f \in \delta(v)$ with $e \neq f$,

we have $\alpha(e) \neq \alpha(f)$?

(i) Let $G = (V, E)$ be an undirected 3-regular graph with a function $\alpha$ satisfying $(*)$. Let $X \subseteq V$ and denote the number of edges $e \in \delta(X)$ with $\alpha(e) = i$ by $n_i$. Show that we have $n_1 \equiv n_2 \equiv n_3 \mod 2$.

(ii) Consider the component $H$ (see Figure 1) as a subset of a 3-regular graph with $a, b, c, d, e$ as its only outgoing edges. Show that for any $\alpha$ satisfying $(*)$, we have

- $\alpha(a) = \alpha(b)$, but $\alpha(c), \alpha(d), \alpha(e)$ are pairwise different, or
- $\alpha(c) = \alpha(d)$, but $\alpha(a), \alpha(b), \alpha(e)$ are pairwise different.

(iii) Let $t \in \mathbb{N}, t \geq 2$. Use $2t$ copies of $H$ to construct a subset $H_t$ of a 3-regular graph with a partition of $\delta(H_t)$ into $t$ pairs of edges $(e_i, f_i), i = 1, \ldots, t$, s.t. for any $\alpha$ satisfying $(*)$, we have either $\alpha(e_i) = \alpha(f_i)$ for all $i$ or $\alpha(e_i) \neq \alpha(f_i)$ for all $i$.

(iv) Consider the component $K$ (see Figure 2) as a subset of a 3-regular graph. Show that for any $\alpha$ satisfying $(*)$, at least one of the pairs $(e_i, f_i)$ of outgoing edges fulfills $\alpha(e_i) = \alpha(f_i)$.

(v) Prove the NP-completeness of the problem above.

Hint: Use a reduction of 3SAT and the components defined in (ii) - (iv). Associate a variable $x$ with a copy of the component $H_t$, where $t$ is the number of occurrences of $x$ or $\overline{x}$ in any clause.

(2+2+2+2+2 points)
Exercise 2.2:
Consider the following algorithm for the MAXIMUM CUT problem: Given an undirected graph $G = (V, E)$, find a set $X \subseteq V$ maximizing $|\delta(X)|$. Start with $X = \emptyset$. If adding a single vertex to $X$ or deleting a single vertex from $X$ makes $|\delta(X)|$ larger, then do so. Repeat until no improvement is possible.

(i) Show that the algorithm is a $\frac{1}{2}$-factor approximation algorithm.

(ii) Does the algorithm always find an optimum solution for planar graphs, or for bipartite graphs?

(4+2 points)

Please turn in your solutions on Tuesday, April 28th, before the lecture.