Problem 1. Consider a complete bipartite graph $G = (V, E)$:
- $V$ has $2n$ vertices, including $n$ black vertices and $n$ white vertices.
- $E$ has $n^2$ edges, including an edge between every black vertex and every white vertex.

Use $G$ to explain why 2 is the best the approximation ratio that we can prove for the vertex cover algorithm discussed in our lecture.

Problem 2*. Let $G = (V, E)$ be an input graph to the vertex cover problem. If $G$ is a tree, describe an $O(|V|)$-time algorithm that finds an optimal vertex cover of $G$.

(Hint: Dynamic programming.)

Problem 3**. Prof. Goofy proposes the following algorithm to find a vertex cover of $G = (V, E)$:

\begin{algorithm}
\textbf{algorithm} max-deg-VC
\begin{enumerate}
  \item $S = \emptyset$
  \item while $E$ not empty do
  \item \hspace{1em} $v \leftarrow$ a vertex with the maximum degree in the current $G$
  \item \hspace{1em} add $v$ to $S$
  \item \hspace{1em} remove from $E$ all the edges of $v$
\end{enumerate}
\end{algorithm}

Show that the approximation ratio of this algorithm is greater than 2.

Problem* 4 (Max-Cut). Let $G = (V, E)$ be a simple undirected graph. Given a subset $S \subseteq V$, a cut induced by $S$ is the set of edges $e \in E$ such that $e$ has a vertex in $S$ and another vertex in $V \setminus S$. Let $\text{OPT}_G$ be the maximum size of a cut that can be induced by any $S \subseteq V$. Design a poly($|V|$)-time (i.e., polynomial time in $|V|$) algorithm that returns a cut of size at least $\text{OPT}_G/2$ in expectation.

(Hint: Random assignment.)