**MAT-72006 Advanced Algorithms and Data Structures**
**December 1, 2016**
**HW 11: 31 Number-Theoretic Algorithms, 35 Approximation Algorithms**

# 1.

Suppose we have a problem we can solve in polynomial time. Does it make sense to consider an approximation algorithm for such a problem? Why or why not?

# 2.

Consider the following questions.

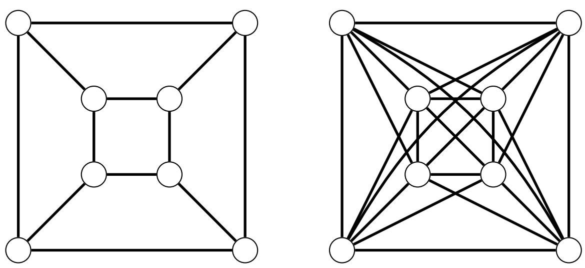
**(a)** Give an example of a graph for which Approx-Vertex-Cover always yields a suboptimal solution.

**(b)** Why is it important to find tight examples for approximation algorithms? (Hint: For example, suppose we have a 2-approximation algorithm for some problem. Why do we care about finding an example where the algorithm returns a solution that is exactly twice as costly as the optimal solution?)

# 3.

Let $G = (V, E)$ be an undirected graph. A *$k$-coloring* of $G$ is an assignment of $k$ colors to its vertices (a $k$-coloring is also known as a *vertex-coloring* if $k$ is not specified). A $k$-coloring is said to be *proper* if no two adjacent vertices have the same color. The minimum $k$ for which the graph $G$ has a proper $k$-coloring is known as its *chromatic number*, denoted by $\chi(G)$.

Consider then the following graphs depicted below as **(a)** and **(b)**. The graph **(b)** is constructed from graph **(a)** by adding an edge between two vertices if their distance is two in **(a)**. Determine the chromatic numbers of the graphs **(a)** and **(b)**.



(a) (b)

## 4.

The *degree* of a vertex is the number of edges incident to it (for example, the degree of each vertex of graph **(a)** of Problem 3 is 3). There is a simple greedy algorithm that finds a proper vertex-coloring for a graph $G$ using $\Delta + 1$ colors, where $\Delta$ is the maximum degree of the graph, i.e., the largest degree over all vertices.

Give a tight example for this algorithm, i.e., prove that there is a graph $G$ such that $\chi(G) = \Delta + 1$.

## 5.

In the *maximum clique problem*, we are given a graph $G = (V, E)$ and an integer $k$. The goal is to decide if there is a clique on $k$ vertices, that is, a complete subgraph on $k$ vertices.

Give a polynomial-time algorithm for solving the maximum clique problem, when the input graph is *a tree*.