1. In class, we defined the notion of approximation ratio for MINimization problems, so that the ratio is a real function >= 1.
   Give a precise definition of approximation ratio for MAXimization problems, so that the ratio is ALSO a real function >= 1.

2. Consider the following algorithm to find a small vertex cover in an undirected graph G:

   > Start with every vertex in C, then for each vertex in turn, remove it from C if C - v still covers every edge (G itself is unchanged).

   Show that this algorithm has a non-constant approximation ratio. What is the worst-case running time for this algorithm?

3. (a) Write a natural greedy algorithm to find a small vertex cover in an undirected graph G. (Hint: think about the degree of each vertex.)

   (b) What is the worst-case running time of your algorithm?

   (c) Show that the approximation ratio for your algorithm is not constant. Warning: this is tricky! Here is a hint: start with a number of disjoint edges, then add new vertices that will be connected to one endpoint of these edges in groups of 3, then groups of 4, ..., up to one last new vertex that connects to each edge. And remember that \sum_{i=1}^{oo} 1/i is Theta(log n).