- This exam has 4 questions for a total of 100 marks.
- Use extra sheets to do your rough work. Minimize rough work on the answer sheets.
- You will **not** get the credit for an answer if you do one of the following:
  - Not clearly state all results that you use;
  - Use arguments with implicit assumptions;
  - Use unclear or incomplete reasoning, or appeals to intuition;
  - Use "weasel words" such as "clearly" and "obviously".
- It is OK to cite results that were stated and/or proven in class, and (parts of) existing answers from your own answer sheet. You may also freely use well-known properties of mathematical objects, but you must **clearly state** what it is that you use.
- **You may get partial credit for answers which are partly correct.** So you should consider writing down what you know to be correct, even in those cases where you cannot provide a proof.

1. Recall that a *vertex cover* of a graph G is a subset $S \subseteq V(G)$ of vertices of G such that every edge in G has at least one of its two end-vertices in S. A set $S \subseteq V(G)$ is said to be a *minimal* vertex cover of G if (i) S is a vertex cover of G, and (ii) no *proper* subset $S' \subsetneq S$ of S is a vertex cover of G. A set $S \subseteq V(G)$ is said to be a *minimum* vertex cover of G if (i) S is a vertex cover of G, and (ii) there is no vertex cover S' of G such that $|S'| < |S|$. Give an example which shows that a mini*mal* vertex cover of a graph G need not be a mini*mum* vertex cover of G. [5]

2. Recall the parameterized VERTEX COVER problem from class:

VERTEX COVER	**Parameter:** k
Input: A graph G and a positive integer k.	
Question: Does G have a vertex cover of size at most k?	

Recall the simple 2-way branching algorithm for VERTEX COVER: while the number of "already chosen" vertices is less than k, pick a remaining edge and branch on its two end-vertices.

Observe that one can naturally associate each node v of the *branching tree* of this algorithm with a set of at most k vertices of the input graph G. These are the vertices which the algorithm has—by the time its execution reaches node v of the *branching tree*—already added to the vertex cover which it is trying to build, on its current path of execution.

In particular, graph G has a vertex cover of size at most k if and only if there is at least one *leaf node* L of the *branching tree* such that the set associated with L is a vertex cover of G.

(a) Prove that for any *minimal* vertex cover (see Question 1 for the definition) S of G of size at most k, there exists *some* leaf node of the branching tree of the above algorithm such that the set associated with this leaf node is *exactly* S. [15]

(b) Prove that for any graph G the number of minimal vertex covers of G of size at most k is at most 2^k. [10]