Exercise 1. In this problem we develop a factor-2 approximation algorithm for the Vertex Cover problem (due to Bar-Yehuda and Even). Without loss of generality assume that the input graph G is connected. The algorithm picks an arbitrary vertex r, constructs a depth-first search tree T rooted at r and outputs the set I of the internal (i.e., non-leaf) vertices of T.

1. Show that the algorithm indeed outputs a vertex cover.

   Hint: Show that the leaves of T form an independent set.

2. Show that if T has a matching of size at least $|I|/2$ then this algorithm gives a factor-2 approximation.

3. Show that in every rooted tree the number of leaves is given by

$$1 + \sum_{v \in I} (d_v - 1),$$

   where I is the set of internal vertices and d_v is the number of the children of v.

4. Fix a maximal matching M of T. Call a vertex (which is not the root) *free* if it is unmatched and its parent is matched with a child. Then M is called a *nice* matching if all the unmatched vertices of T (other than the root) are free. Show that a nice maximal matching alway exist.

   Hint: Show that every unmatched vertex at the level $k > 0$ (with the root-level being defined as zero) that is not free can be fixed using local modifications. Then, starting from the highest level, inductively *push out* all unmatched vertices that are not free towards the root.

5. Fix a nice matching M of T. Show that the number of leaves of T is lower bounded by the number of unmatched vertices.

6. Conclude that T has a nice matching of size at least $|I|/2$, and hence, the algorithm gives an approximation guarantee of 2.

Exercise 2. The goal here is to find another proof that the greedy algorithm given in the course for the Vertex Cover problem is a factor-$O(\log m)$ approximation.

- If we have a cover of size OPT, what does it imply on the maximum degree of a vertex in the graph?
- We recall that the greedy algorithm adds at each step the vertex of maximum degree in the yet uncovered graph. Each time we add a vertex, give an upper bound on the number of uncovered edges so far (Hint : use the first question).
- Deduce that the greedy algorithm is a factor-$O(\log m)$ approximation algorithm.

Exercise 3. In the problem MAX-Ek-SAT we are given a boolean formula in conjunctive normal form, with each clause containing exactly k distinct literals, and the goal is to find the maximum number of clauses that can be simultaneously satisfied (that is, the maximum, over all possible assignments of variables, of the number of satisfied clauses). This problem is NP-hard even for $k = 2$. Show that it is always possible to find a assignement such that at least $m(2^k - 1)/2^k$ clauses are satisfied, where m is the number of clause in the formula.
Remark: In fact a deterministic and efficient algorithm for *finding* an assignment that satisfies the aforementioned bound is known. It turns out that, unless P = NP, this is the best possible polynomial time approximation!

Exercise 4. Find a polynomial algorithm to decide $2 - SAT$.

Exercise 5 (Maximum acyclic subgraph)**.** Given a directed graph $G = (V, E)$, the problem of finding an acyclic subgraph $G' = (V, E')$ with $E' \subseteq E$ and E' of maximum possible cardinality is NP-hard.

- Show that if $G' = (V, E')$ is a maximal acyclic subgraph (that is, we cannot add any more edges to E') then $G'' = (V, E - E')$ is also acyclic.
- Deduce from (1) a factor-2 polynomial approximation to the maximum acyclic subgraph problem.
- What about the case of an undirected graph ?