20. Consider the following algorithms that have as input a boolean formula F in 3-CNF and returns a number between 0 and the number of clauses.

AlgA(F)

- Let c_0 be the number of true clauses when all variables are set to value 0.
- Let c_1 be the number of true clauses when all variables are set to value 1.
- return $\max\{c_0, c_1\}$.

AlgB(F)

- For each variable, toss a fair coin to determine its value, 0 or 1.
- return the number of true clauses under this random assignment.

AlgC(F)

- return $\max\{\text{AlgA}(F), \text{AlgB}(F)\}$.

Prove that the three algorithms are polynomial time (randomized) approximation algorithms for the MaxSAT problem. Which algorithm provides the best ratio?

21. Consider the Max Clique problem. Given an undirected graph $G = (V, E)$ compute a set of vertices that induce a complete subgraph with maximum size.

For each $k \geq 1$, define G^k to be the undirected graph (V^k, E^k), where V^k is the set of all ordered k-tuples of vertices from V. E^k is defined so that $(v_1, \ldots, v_k)$ is adjacent to $(u_1, \ldots, u_k)$ iff, for $1 \leq i \leq k$, either $(v_i, u_i) \in E$ or $v_i = u_i$.

- (a) Prove that the size of a maximum size clique in G^k is the k-th power of the corresponding size in G.
- (b) Argue that if there is a constant approximation algorithm for Max Clique, then there is a polynomial time approximation schema for the problem.

22. Consider the Max Cut problem: given an undirected graph $G = (V, E)$, find a set $S \subseteq V$ such that $|\{(u, v) \in E \mid u \in S, v \notin S\}|$ is máximum.

Consider the greedy algorithm that starts placing an arbitrary vertex in S and then proceed over the remaining vertices in V in some order. A vertex v is added to S if $N(u) \cap S < N(u) \cap V \setminus S$.

Show that the proposed algorithm is a polynomial 2-approximation for Max Cut.

23. Consider the Graph without triangles problem: given an undirected graph $G = (V, E)$, find the a set $E' \subseteq E$ with maximum size such that $G' = (V, E')$ does not contain any triangle, i.e., in G' you cannot find vertices x, y, z with $(x, y), (x, z), (y, z) \in E'$.

The problem is NP-hard. Design a 2-approximation algorithm for the problem and provide its cost.