Problem 1. Consider the following Max-3-Sat problem.
Instance: A formula φ in 3-CNF.
Feasible solution: An assignment v of truth values to variables.
Objective: Maximize the number of satisfied clauses.

- Construct polynomial time approximation algorithm with error 2.
- Prove that there exists an assignment satisfying $\frac{7}{8}$ clauses.
- Construct an approximation algorithm with error $\frac{1}{0.74}$ and polynomial expected running time.

Problem 2. How hard is it to decide for a given formula φ the following questions if φ is a general formula or CNF or DNF.

- $\exists v : \varphi(v) = 0$
- $\exists v : \varphi(v) = 1$
- $\forall v : \varphi(v) = 0$
- $\forall v : \varphi(v) = 1$

Problem 3 (3 points)**.** Consider the following parametrized problem.
Parameter: $k \in \mathbb{N}$
Instance: A set of n points in a plane.
Question: Are there k lines covering all points?
Construct FPT algorithm and kernelization.

Problem 4 (1 point)**.** Decide whether the following problem is polynomial or NP-complete.
Instance CNF formula ϕ
Question Are there two different assignments u, v of truth values to variables in ϕ which satisfy ϕ, i.e. $\phi(u)$ and $\phi(v)$ both evaluate to true.

Problem 5 (2 points)**.** Construct a Turing machine which recognizes language $L = \left\{0^{(2^n)};\ n \in \mathbb{N}\right\}$, i.e. words containing only zeros whose length is a power of two. Write precisely your Turing machine!

Problem 6 (4 points)**.** For each pair of complexity classes below try to decide if we can prove a relation between them (by using theorems and propositions from the lecture). That is, for every pair of classes A and B decides whether $A \subseteq B$, $B \subseteq A$, $A \setminus B = \emptyset$ and $B \setminus A = \emptyset$. Note that some relations may not be known.

1. $\mathsf{DSPACE}(n^3)$ and $\mathsf{DTIME}(2^{n^3})$
2. $\mathsf{DTIME}(2^{n^3})$ and $\mathsf{NSPACE}(n \log n)$
3. $\mathsf{NSPACE}(n \log n)$ and $\mathsf{NTIME}(n \log n)$
4. $\mathsf{NTIME}(n \log n)$ and $\mathsf{DSPACE}(n)$
5. $\mathsf{NTIME}(n \log n)$ and $\mathsf{DSPACE}(n^3)$

Problem 7 (4 points)**.**

- Prove that one-tape Turing machines which are allowed to write on every cell at most twice are equivalent to (standard) Turing machines.
- Prove that one-tape Turing machines which are allowed to write on every cell at most once are equivalent to (standard) Turing machines.