Things that have been added compared to the earlier version of this guide are in red.

The exam will be open book / open notes / open screenshot (but no laptops or other electronic/communicating devices).

You should be able to...

- ...give runtime bounds for simple algorithms in $O$-notation.
- ...design Turing machines for (very simple) algorithms.
- ...use the Church-Turing thesis to justify the use of Turing machines in complexity theory.
- ...explain the consequences of the existence of the halting problem.
- ...apply universal Turing machines in simple reasonings about Turing machines.
- ...given simple problems, show that they are in $\text{DTIME}(n^c)$ for certain $c$.
- ...given simple problems, show that they are in $\text{P}$.
- ...identify problems that are not in $\text{P}$. (E.g., when presented with several problems.)
- ...use the Strong Church-Turing thesis to justify the use of the class $\text{P}$ in complexity theory.
- ...explain the intuitive meaning of the class $\text{NP}$.
- ...for each of the parts of the definition of $\text{NP}$, explain why that part needs to be the way it is. (I.e., what would happen if we change the definition, e.g., do not restrict the witness to be polynomial-length.)
- ...given various problems, identify those which are in $\text{NP}$, and which are probably not in $\text{NP}$ (and give justification).
- ...given two languages, construct a polynomial-time Karp reduction between them (especially important: get the direction right).
- ...show that a given language is $\text{NP}$-hard / $\text{NP}$-complete.
- ...explain under which conditions $\text{NP}$-hardness tells us that problems have no efficient algorithms (and in which sense? deterministic? randomized? etc.)
• ...sketch how “SAT is NP-hard” is proven. (On a high level, focusing on what is transformed into what.)
• ...distinguish NP and coNP. (E.g., tell which problems are in which class.)
• ...given hardness of a decision problem, show the hardness of a search problem, and vice versa.
• ...write oracle algorithms that, given an oracle for a certain language $L$, decides a certain language $L'$.
• ...explain the significance of Baker-Gill-Solovay for how difficult it is to decide whether $P = NP$ or $P \neq NP$.
• ...explain why each part of the definition of BPP is the way it is. (I.e., what would change or not change if some parts of the definition would be changed? E.g., changing the numbers from the success probabilities.)
• ...give a sketch why amplification in BPP is possible. (I.e., why the error probability can be reduced.)
• ...explain the differences between BPP, RP, coRP, ZPP.
• ...explain the relation of BPP and the Strong Church-Turing thesis and how the Strong Church-Turing thesis has to be amended if $P \neq BPP$.
• ...identify whether a given algorithm is a PSPACE-algorithm or not.
• ...identify languages that are in PSPACE.
• ...given a simple two-player game, write down a PSPACE-algorithm for it.
• ...explain why each part of the definition of $\Sigma^p_i, \Pi^p_i, \text{PH}$ is the way it is. (The same for the alternative definitions of $\Sigma^p_i, \Pi^p_i$.)
• ...identify which $P, NP, coNP, \Sigma^p_i, \Pi^p_i, \text{PSPACE}$ are related to each other in which way? (Inclusion, inequality, under which conditions?, what happens if they are included in each other?)
• ...identify $\Sigma^p_i, \Pi^p_i$-complete problems.
• ...for simple problems, draw a circuit.
• ...given a circuit, identify properties such as whether it is a valid circuit, fan-in, depth, width, number of variables.
• ...explain why each part of the definition of $P/poly$ is the way it is.
• ...explain why $P$ and $P/poly$ are different classes. (Both by giving separating languages, and by giving the intuitive reason what makes $P/poly$ stronger.)
• ...recognize problems that are in $P/poly$ (in simple cases).
• ...explain why $P_{/\text{poly}}$ is still a relevant class to study in complexity theory.

• ...explain why each part of the definition of $\text{MA, IP}$ is the way it is.

• ...identify languages which are in $\text{MA or IP}$. (Assuming they are similar in concept to the languages we saw in the lecture to be contained in $\text{MA, IP}$.)

• ...evaluate a quantum circuit (i.e., what is the final state, what is the probability of a given measurement outcome).

• ...explain why each part of the definition of $\text{BQP}$ is the way it is.

• ...explain the significance of Simon’s algorithm on the question whether $\text{BQP}$ is larger than $\text{BPP}$. (And the reason why the existence of Simon’s algorithm does not prove that $\text{BQP}$ is larger than $\text{BPP}$.)

• ...recognize whether a function is negligible or not (in simple cases).

• ...recognize whether a function is a one-way function or not.

• ...recognize whether an encryption scheme satisfies the definition of secrecy (IND-OT-CPA) (in simple cases, or for schemes similar to ones seen in lecture/practice/homework).

• ...explain what a PCP (“probabilistically checkable proof”) is. (I.e., explain the intuition what a PCP achieves.)

• ...explain why each part of the definition of $\text{PCP / PCP-verifier}$ is the way it is.

• ...construct a PCP for a given language. (Assuming it is similar in concept to the languages we saw in the lecture to have a PCP.)

• ...explain how the PCP-theorem (“$\text{NP = PCP}(\log(n), 1)$”) implies that qCSP is hard to approximate. (I.e., you should be able to get right what is reduced to what and how a PCP can be seen as a qCSP instance, and how that all fits together to imply hardness of approximation of qCSP. And under which additional complexity theoretic assumptions?)

• ...for any two complexity classes, tell how they are related (e.g., inclusion, equality, non-empty intersection, incomparable, etc.), possibly under common assumptions (like $P \neq \text{NP, PH}$ does not collapse, etc.). (Only if the corresponding information was given in the lecture or practice. Even though this seems to be something you can just look up in the notes when a question comes, I recommend to study for this anyway, because it may take a lot of time to find the relation for a given pair of classes when needed in the exam. Although this study guide point looks like it is just one of many points, it is basically an umbrella covering many different chapters of the lecture.)

• Good luck!