# Learning Outcomes for Computational Complexity Theory

## 1 Definition Specific Goals

1. Define TM.
2. Define configuration of a TM.
3. Define halting and looping.
4. Define TMs in detail for simple algorithms.
5. Describe TMs for complex algorithms.
6. Prove equivalence of various definitions of TMs.
7. Define time complexity.
8. Prove the time hierarchy theorem.
9. Define $\mathsf{P}$.
10. Define $\mathsf{NP}$ using verifiers
11. Define $\mathsf{NP}$ using non-deterministic TMs.
12. Define $\mathsf{coNP}$ using refuters
13. Define $\mathsf{coNP}$ using co-non-deterministic TMs.
14. Define $\leq_p^m$.
15. Define $\leq_p$ via oracle TMs.
16. Prove $\mathsf{NP}$-completeness using gadgets.
17. Define $\mathsf{EXP}$.
18. Define $\mathsf{NEXP}$.
19. Translate separations using padding arguments.
20. Define space, read-only tape.
21. Prove the space hierarchy theorem.
22. Define $\mathsf{PSPACE}$.
23. Define $\mathsf{L}$.
24. Define $\mathsf{NL}$.
25. Define $\mathsf{coNL}$.
26. Define log-space transducers.
27. Define $\leq_{\log}^m$.
28. Define $\leq_{\log}$.
29. Prove $\mathsf{NL} = \mathsf{coNL}$.
30. Define $\Sigma_i^\mathsf{P}$ using complete problems.
31. Define $\Pi_i^\mathsf{P}$ using complete problems.
32. Define $\mathsf{PH}$.
33. Define Boolean circuits.
34. Define Boolean circuit size.
35. Define $\mathsf{P/poly}$.
36. Prove Karp-Lipton theorem.
37. Define randomized TMs.
38. Define $\mathsf{BPP}$.
39. Define $\mathsf{RP}$.
40. Define $\mathsf{coRP}$.
41. Define $\mathsf{ZPP}$.
42. Prove that polynomial identity testing is in $\mathsf{BPP}$.
43. Prove that finding perfect matchings is in $\mathsf{RP}$.
44. Prove $\mathsf{RP} \cap \mathsf{coRP} = \mathsf{ZPP}$.
45. Prove Adelman's theorem.
46. Define $\#\mathsf{P}$.
47. Prove Toda's theorem.
48. Define $\mathsf{IP}$.
49. Prove $\mathsf{IP} = \mathsf{PSPACE}$.
50. Define Boolean circuit depth.
51. Define $\mathsf{NC}^i$ for $i \geq 0$.
52. Define $\mathsf{AC}^i$ for $i \geq 0$.
53. Prove $\mathsf{AC}^i \subseteq \mathsf{NC}^{i+1}$ for $i \geq 0$.
54. Prove $\mathsf{NL} \subseteq \mathsf{NC}^2$.
55. Prove $\mathsf{NC}^1 \neq \mathsf{AC}^0$.
56. Define branching programs.
57. Define non-deterministic branching programs.
58. Prove equivalence of b.p. size and space.
59. Define layered branching programs.
60. Define width of layered b.p.
61. Prove equivalence of b.p. and layered b.p.
62. Prove Barrington's theorem.

## 2 Cross-cutting Goals

1. For each definition, identify examples and non-examples.
2. Prove membership of well-known problems in classes.
3. Prove well-known containment relationships.
4. Prove well-known closure wrt set operations for classes.
5. Prove well-known closure for classes wrt reductions.
6. Choose suitable reductions to define hardness based on the complexity class $\mathcal{C}$.
7. Identify canonical complete problems for classes.
8. Prove completeness for well-known problems.