Reductions

- We used an algorithm $\mathcal{A}$ for decision problem π to solve an optimization or construction problem π' on the same input. We also required that we call $\mathcal{A}$ at most a polynomial number of times, and that we do at most a polynomial number of other operations.
- This means that if $\mathcal{A}$ runs in polynomial time, then we have a polynomial time algorithm for both π and π'. Note that we use two things here: $\mathcal{A}$ is polynomial, and the input did not change in size.
- What we did isn't really a Karp reduction, because Karp reductions are only for decision problems... but the ideas are very related.
- If you can understand why this works, you will understand why Karp reductions have to satisfy what they satisfy.

Just try to understand the ideas. This is not about memorization.