BOUNDS
(as guarantees, no a priori distribution on inputs)

DETERMINISTIC

**worst case**
- unreferenced with respect to OPT, non-amortized

**approximation**
- embed OPT within (constant) factor to algorithm, in worst case
- correctness proof often gives a factor to OPT solution (e.g. solution to NP-hard problem)
- competitive analysis gives a factor to OPT runtime (off-line)

**amortized**
- average over sequence of operations, in worst case

PROBABILISTIC

**expectation**
- get a probability bound with Markov inequality etc..

**approximation in expectation**
- embed OPT within a (constant) factor to algorithm, in expectation

**with high probability**
- error probability is polynomially small
- event description can have a bound; given an O(g(n)) event bound, error probability is an upper bound on probability of exceeding the event bound

A randomized algorithm can be a starting point for designing a deterministic/derandomized algorithm (from expectation to worst-case bounds), and vice versa (e.g. keep correctness and improve time/space). Randomization does not seem to solve NP-complete problems in polynomial time :)