Probabilistic Algorithms
One Example

Matrix multiplication:

- A,B,C: $n \times n$ matrices.

- Computing AB takes $O(n^3)$ time. Thus deterministically check if $AB = C$ takes $O(n^3)$ time.

- However, probabilistically we can do better.

We can check $AB = C$ in $O(n^2)$ time with high probability.

Algorithm (Check if AB = C):

```
Repeat k times:
   Generate a random n x 1 vector with
   entries from {-1,1};
   Test if A(Bx) = Cx.
      if not, then AB != C.
If we pass above loop, output AB = C.
```

Theorem: (1) The above algorithm errs with probability $< 2^{-k}$; (2) The time compleixty is $O(n^2)$.

Proof. (2) is trivial. We prove (1).

$ABx = Cx$? If $\sum_{\ell=1}^{n} a_{i\ell} b_{\ell j} \neq C_{ij}$, then $ABx \neq Cx$ for at least one value of $x_j = 1$ or $x_j = -1$.

Thus with probability 1/2, one loop will tell if $AB \neq C$. After k loops, we have probability less than $(1/2)^k$ to fail. □