Description: Suppose the running time of the randomized algorithm $A$, given an input of size $n$ is $O(n)$ and that it generates the correct output with probability $1/n$.

1. How many times do we need to run algorithm $A$ so that the probability that “all runs have failed” is smaller that $\epsilon$.

Suppose we need to run algorithm $k$ times. The probability that all $k$ runs of algorithm $A$ failed is $(1 - \frac{1}{n})^k$. It is sufficient to set $k \geq \frac{\log \epsilon}{\log(1 - \frac{1}{n})}$ such that $(1 - \frac{1}{n})^k \leq \epsilon$.

2. What is the expected number of times $A$ is run until the first successful run?

Let $E$ be the expected number of times $A$ is run until the first successful run. We have the following recurrence relation.

$$E = \Pr(A \text{ is failed}) \cdot (E + 1) + \Pr(A \text{ is succussed}) \cdot 1$$

$$= (1 - \frac{1}{n}) \cdot (E + 1) + \frac{1}{n} \cdot 1.$$

Solving the above equality gives us $E = n$.

3. Suppose that $B$ can check whether an input-output is correct in time $O(n)$. What is the expected running time for solving the problem using $A$ and $B$ as a function of $n$.

Again, let $E$ be the expected running time. We also assume the running time is at most $cn = O(n)$ where $c$ is positive constant. We have the following recurrence relation.

$$E \leq \Pr(A \text{ is failed}) \cdot (E + cn) + \Pr(A \text{ is succussed}) \cdot cn$$

$$\leq (1 - \frac{1}{n}) \cdot (E + cn) + \frac{1}{n} \cdot cn.$$

Solving the above equality gives us $E \leq cn^2 = O(n^2)$. 