Advanced Algorithms 2014. Exercises 7-8

Remark: Technically the following exercises should be rather simple. But what we also evaluate is the clean and correct use of probability-theoretic concepts. Show and explain your calculations, not only the final answers. Prove all claims, e.g., about expected numbers.

Exercise 7.
Imagine that $n$ people vote in an election. They have the choice between two candidates $A$ and $B$. Actually $k$ people want to vote for $A$, and $n-k$ people for $B$. However some voters get confused, more precisely: every voter, independently and with probability $p$, votes for the wrong candidate. (Probability $p$ is a fixed number.) What is the expected number of votes for $A$ and $B$, respectively? You may also add a discussion of the result.

Exercise 8.
A set $S$ of $n$ distinct elements is stored in an array, in arbitrary order. We wish to find a specific element $x$. (It is known in advance that $x \in S$, but we do not know the place of $x$.) The only action we can take is to look up elements in the array.

8.1. For trivial reasons, any deterministic algorithm needs $n$ look-ups in the worst case. Give a randomized algorithm that finds $x$ after roughly $n/2$ expected look-ups.

8.2. This was probably(!) not too difficult. However, generating random numbers for randomized algorithms is an issue, and we do not like to waste random bits. Therefore we add a further demand: Give a randomized algorithm that finds $x$ after roughly $n/2$ expected look-ups, and uses only one random bit (flips a coin only once).