Exercise 1.

(i) For any comparison-based algorithm, it's clear that each comparisom can produce a smaller one and a larger one, and the larger one have no chance to be the smallest number. For each number except the smallest one, it needs exactly one comparison to be excluded. Therefore, there are exactly $n-1$ comparisons in such algorithm.

(ii) similar to (i), we can first find the smallest number in $n-1$ comparisons.

Then for the second smallest number, we know that this number must have been compared with the smallest number in the above comparisons, so the possible second smallest number is among $\lceil \log_2 n \rceil$ numbers, which equals to how many times the smallest number had been at least compaired.

Therefore, we can have a new list of $\lceil \log_2 n \rceil$ numbers, and need $\lceil \log_2 n \rceil - 1$ comparisons to find the smallest one which is the second smallest of the original list.

So totally we need $n-1+\lceil \log_2 n \rceil - 1 = n-1+\lfloor \log_2 n \rfloor$ comparisons.