Data Structures and Algorithms

A certain program has to maintain an array, $\text{count}$, of $N$ counters which are all initialised to zero. The value of counter $i$ can be incremented by one by the call: $\text{increment}(i)$, and this is the only way the program changes counter values. Two variables, $\text{mincount}$ and $\text{maxcount}$, must always hold the smallest and largest of the counter values whenever the point of execution is not within the function $\text{increment}$. You may assume that $\text{increment}$ is called about $1000N$ times when the program is run and that its argument is typically uniformly randomly distributed between 1 and $N$, but on some runs it cycles through the numbers 1 to $N$ in order 1000 times.

(a) Describe, in detail, an efficient data structure and algorithm to use when $N$ is expected to be about 10. [5 marks]

(b) Describe, in detail, an alternative data structure and algorithm to use when $N$ is about a million. [10 marks]

(c) Suppose your algorithm for (a) above were used when $N = 10^6$, estimate how much slower it would be compared with your algorithm for (b). [5 marks]