Define an abstract data type *MaxArr (MA)* to hold (possibly repeated) integer values $1, \ldots, n$, with the following operations:
Initialize: $\rightarrow$ *MA* (create an empty MA)
Increment: *MA* $\rightarrow$ *MA* (add a copy of '1' to MA)
Add: *Int* $\times$ *Int* $\times$ *MA* $\rightarrow$ *MA*

(delete a copy of each of the two integers and add a copy of their sum)
DeleteMax: *MA* $\rightarrow$ *MA* (delete a copy of the maximum integer in MA).

Show that when implemented by an array with a "max" pointer, *MaxArr* achieves constant time amortized complexity for all operations.