Algorithms: Solutions 2

(1) Give an algorithm that determines the number of *distinct* elements in an array $A[1..n]$.
(2) Estimate the worst-case running time of your algorithm.

Distinct-Counter(A, n)	*cost*	*times*
counter $\leftarrow 0$	c_1	1
for $j \leftarrow 1$ **to** n	c_2	$n+1$
do $i \leftarrow j+1$	c_3	n
while $i < n+1$ and $A[j] \neq A[i]$	c_4	$\leq \sum_{i=1}^{n}(n-i+1) = \frac{n\cdot(n+1)}{2}$
do $i \leftarrow i+1$	c_5	$\leq \sum_{i=1}^{n}(n-i) = \frac{(n-1)\cdot n}{2}$
if $i = n+1$	c_6	n
then *counter* $\leftarrow$ *counter* $+1$	c_7	$\leq n$
return *counter*	c_8	1

$$\begin{aligned} T(n) &\leq c_1 + c_2\cdot(n+1) + c_3\cdot n + c_4\cdot\frac{n\cdot(n+1)}{2} + c_5\cdot\frac{(n-1)\cdot n}{2} + c_6\cdot n + c_7\cdot n + c_8 \\ &= (\frac{c_4}{2}+\frac{c_5}{2})\cdot n^2 + (c_2+c_3+\frac{c_4}{2}-\frac{c_5}{2}+c_6+c_7)\cdot n + (c_1+c_2+c_8) \\ &= \Theta(n^2) \end{aligned}$$