

Complexité d'un algorithme : exemple

2^e question : lequel est le **plus efficace** ?

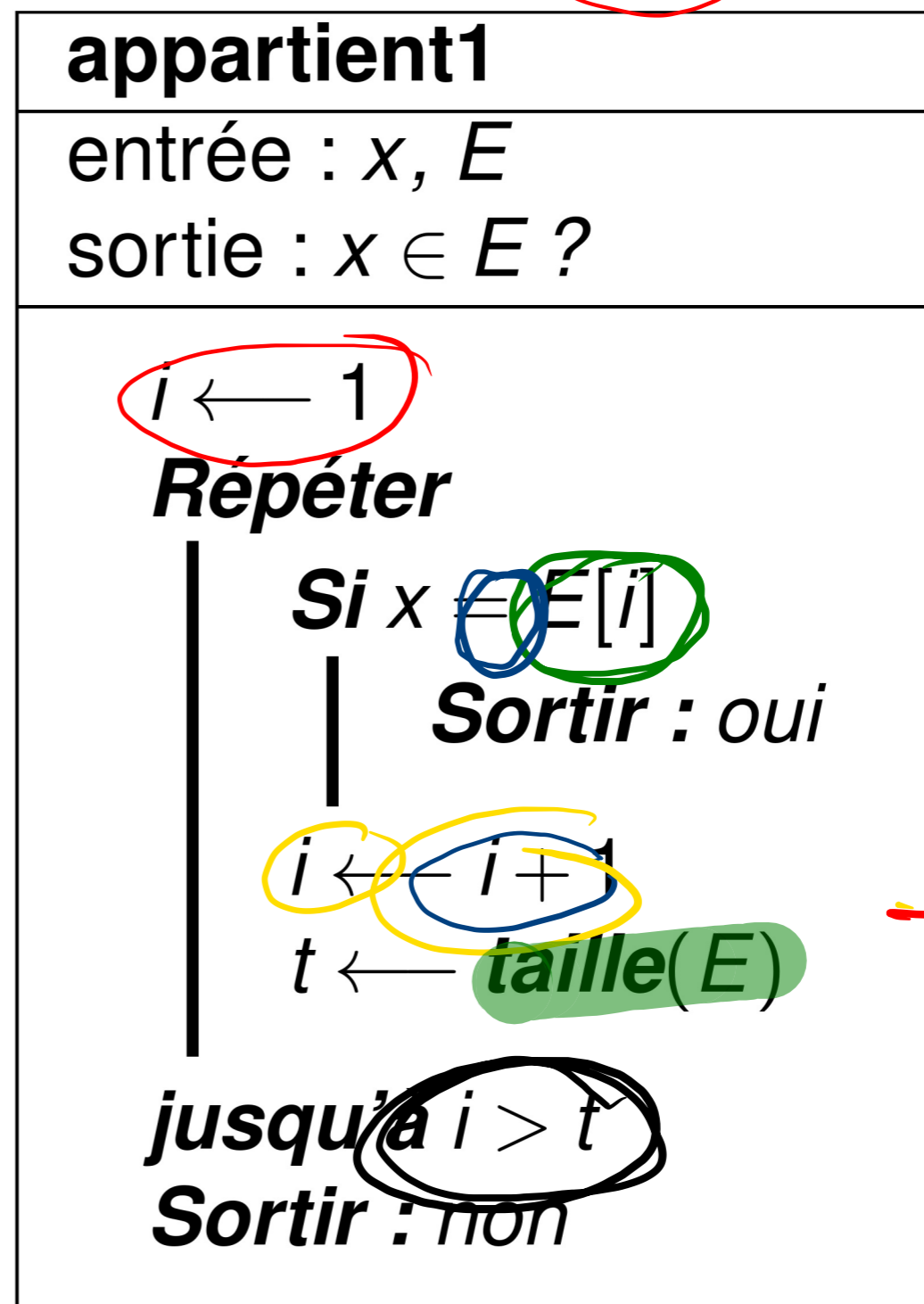
Notons n la taille de E et comptons combien d'instructions élémentaires chaque algorithme nécessite dans le pire des cas $C1(n)$ et $C2(n)$

Pour l'algorithme appartient1(x, E) :

$$C1(n) =$$

↑

$$\boxed{\begin{array}{l} 1 + 1 \\ + 2 \\ + T(n) \\ * 1 \end{array}} \times n$$



$T(n)$

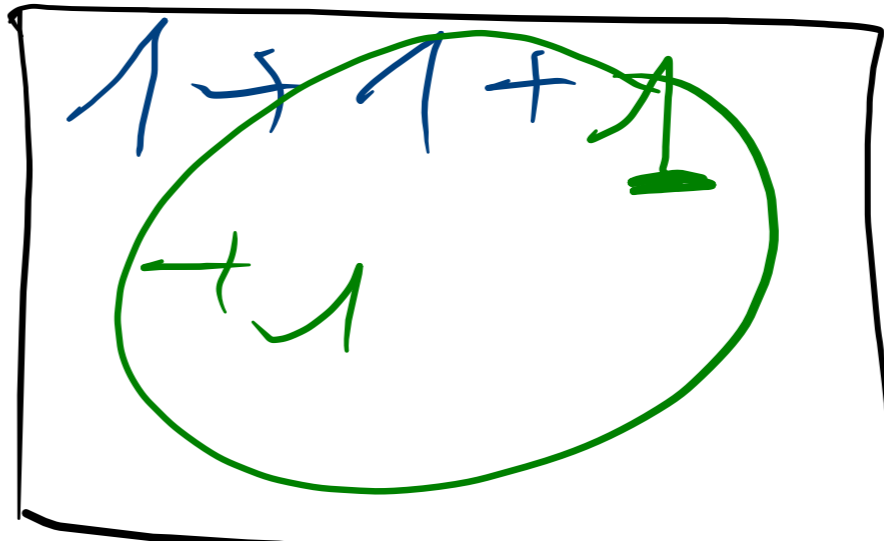
Complexité d'un algorithme : exemple

Pour l'algorithme appartient2(x, E) :

appartient2
entrée : x, E
sortie : $x \in E ?$
<pre> t ← taille(E) Pour i de 1 à t Si x = E[i] Sortir : oui Sortir : non </pre>

$$1 + T(n) + 1$$

$$f n x$$



Constante

Recherche

triée : $O(\log(n))$

non triée : \rightarrow directe $O(n)$

\searrow on trie d'abord
 $O(n \cdot \log n) + O(\log(n)) = O(n \log n)$