* **Glouton :**

- D'abord on trie tous les batons $\Rightarrow \Theta(n \log n)$
- Ensuite on parcourt tous les batons en faisant des opérations de temps constant $\Rightarrow O(n)$

$\Rightarrow$ Total : $\Theta(n \log n)$

* **Progdyn 1 :**

- Premiere boucle $\Rightarrow \Theta(N)$
- Deuxieme boucle $\Rightarrow \Theta(n)$
- Troisieme boucle (avec boucle imbriquée) $\Rightarrow \sum_{j=1}^{N} j/2 = N(N+1)/4$

$\underbrace{1 \longrightarrow N}$ $\underbrace{1 \longrightarrow j/2}$

$\Rightarrow$ Total : $\Theta(n + N^2)$

* **Progdyn 2 :**

1. On utilise un tableau de taille $n \cdot N$ pour stocker nos valeurs $\Rightarrow \Theta(n \cdot N)$
2. Pour chaque valeur on 2 alternatives $\Rightarrow \Theta(1)$

$\Rightarrow$ (1) et (2) donnent Total : $\Theta(n \cdot N)$

* **Recuit :**

- Pour obtenir une solution voisine $\Rightarrow \Theta(n)$
(3 boucles, dont chacune parcourt tous les n batons au pire cas)
- Pour le recuit simulé on a 2 boucles imbriquées

$1 \longrightarrow k_{max}$	$1 \longrightarrow P$	solution voisine	Somme solution
$\Theta(k_{max})$	$\Theta(P)$	$O(n)$	$O(n)$

- Pour initialiser l'algorithme, on a besoin d'une solution gloutonne $\Theta(n \log(n))$
- On a alors : $n \log n + k_{max} \cdot P \cdot n$

$\Rightarrow$ Total : $\Theta(n \log n + k_{max} \cdot P \cdot n)$