

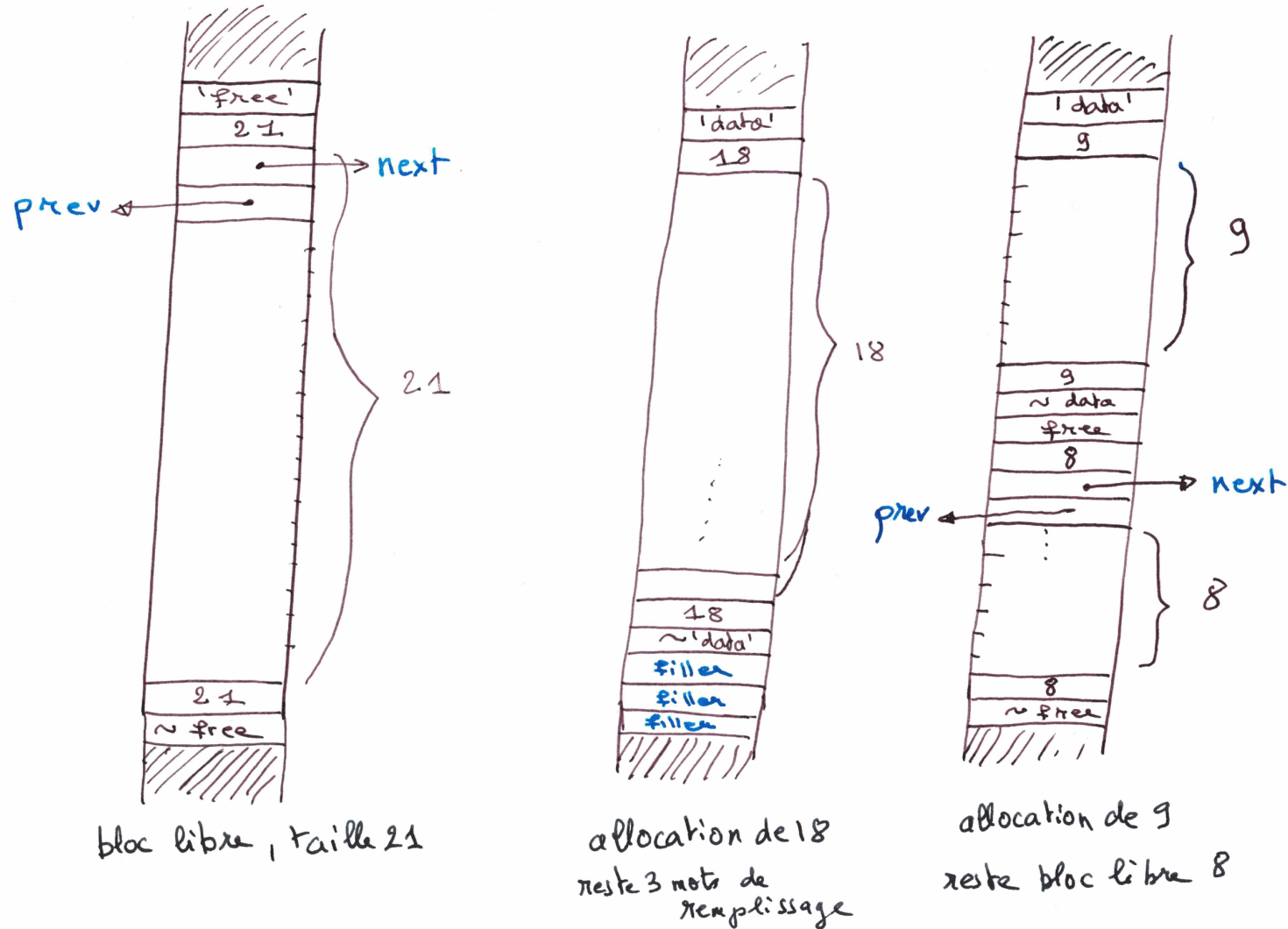
Comment rendre efficace une allocation dynamique ?

- Algorithmes nombreux, anciens, bien connus...
- Besoins « temps réel », durée des opérations strictement bornée : algorithmes en $O(1)$
- Liste des blocs libres doublement chaînée
- Une liste de blocs libres par classe de taille
 - Classes basées sur la série $2/3$
 - 8 (..11) 12 (..15) 16 (..23) 24 (..31), etc.
 - Soit 40 à 50 classes
 - « good fit », perte inférieure à 50% de la taille demandée

Est-ce vraiment en $O(1)$?

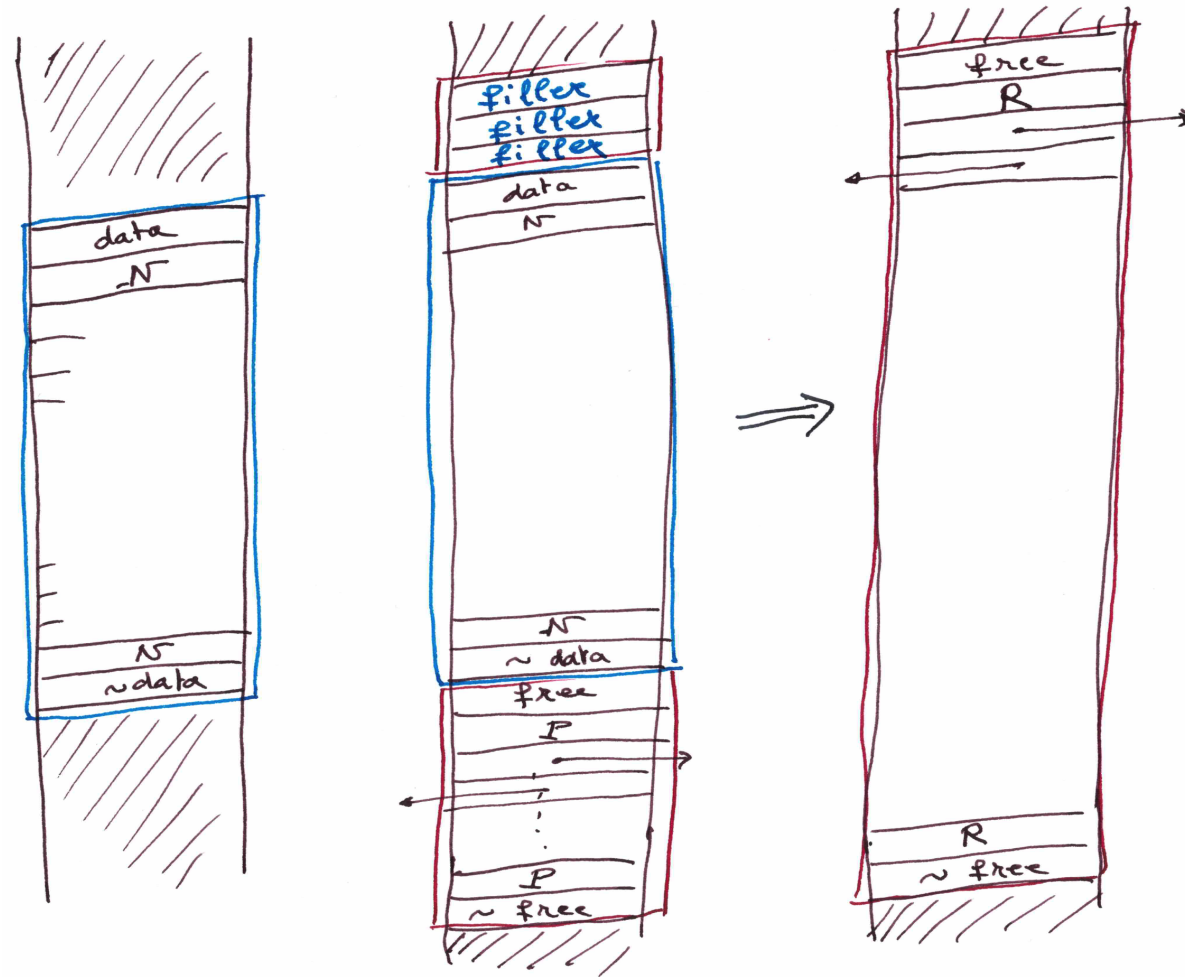
- Pour une demande de « N » mots
 - Recherche dichotomique dans la table des listes
 - $O(\log k)$: $k \leq 50$. Durée strictement limitée
 - Retrait du premier bloc de la liste
 - $O(1)$: 2 manipulations de pointeurs
- Pour une libération d'un bloc de « N » mots
 - Recherche dichotomique dans la table des listes
 - Insertion du bloc libéré en tête (ou en queue)
 - $O(1)$: 2 manipulations de pointeurs

Analyse d'une allocation



Note : il n'y a jamais 2 blocs libres contigus

Analyse d'une libération



libération d'un bloc de mémoire