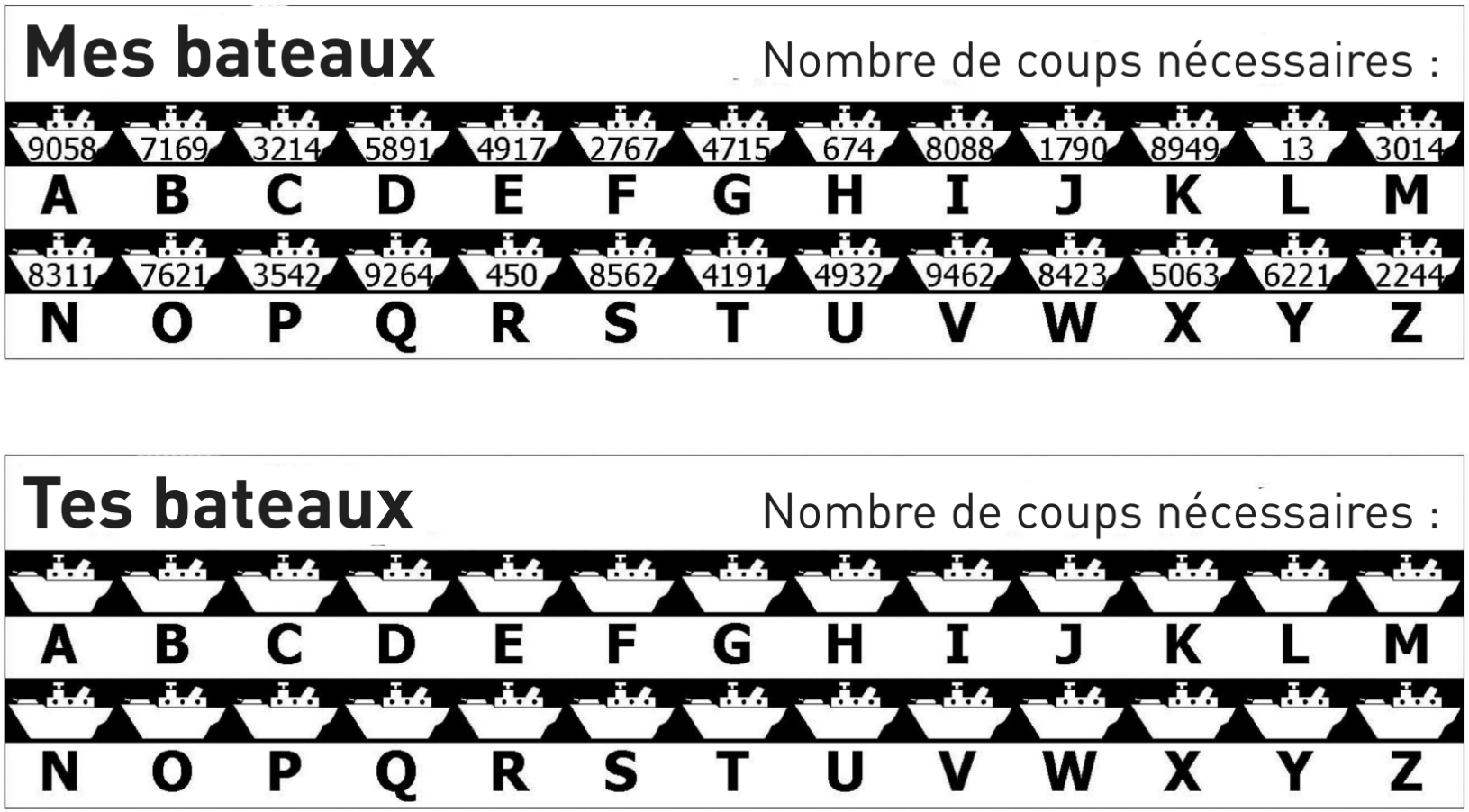
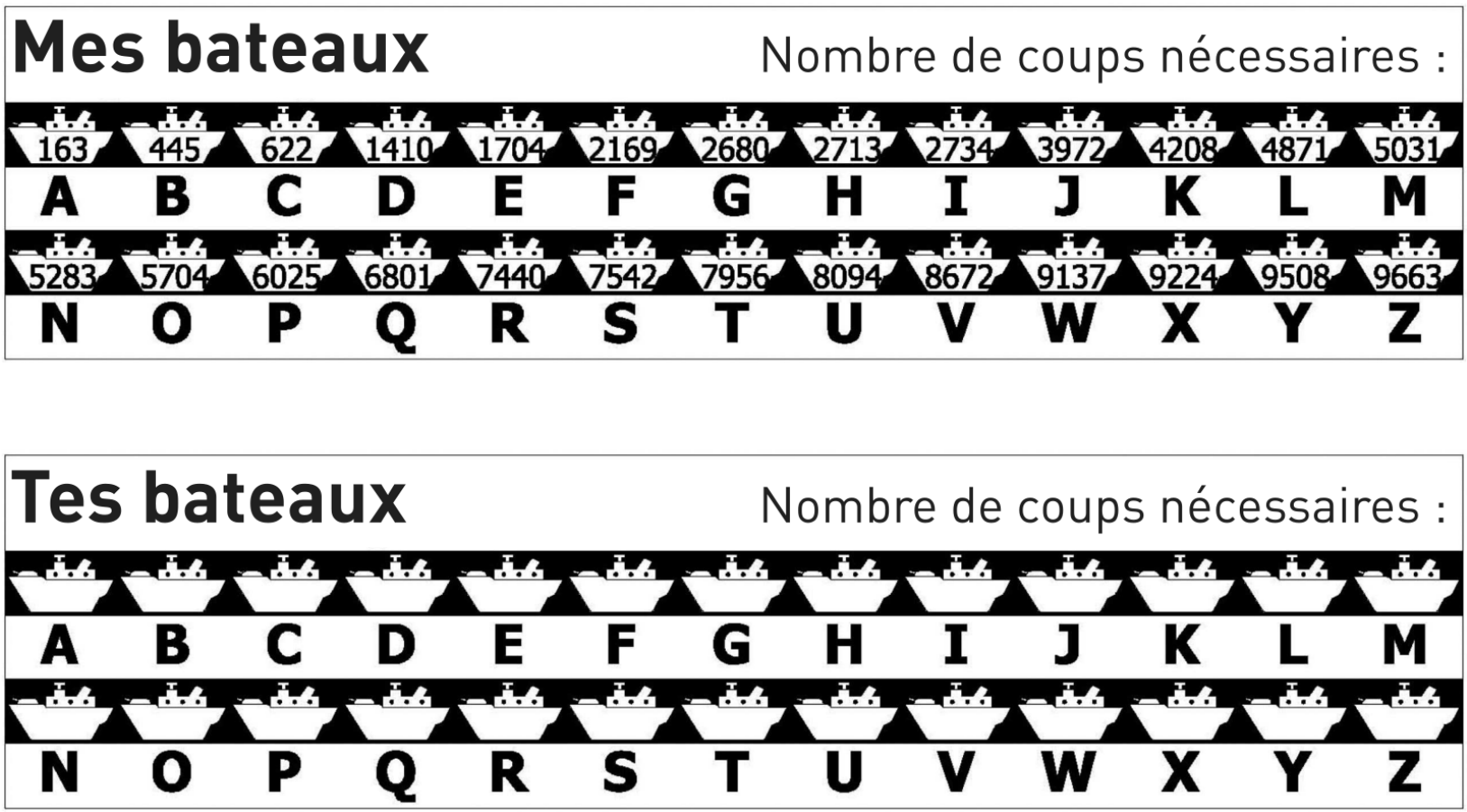
Troisième cours

Structures complexes

Un petit jeu pour les enfants.



Une autre partie ?



Complexité d'un algorithme

Définition :

Complexité en temps : Quantité de temps nécessaire à l'algorithme pour s'exécuter

Complexité en espace : Quantité de mémoire nécessaire à l'algorithme pour s'exécuter.

Complexité d'un algorithme

Définition :

Complexité en temps : Quantité de temps nécessaire à l'algorithme pour s'exécuter

Complexité en espace : Quantité de mémoire nécessaire à l'algorithme pour s'exécuter.

- fortement dépendant de la machine, des éléments utilisés
- plutôt en terme d'ordre de grandeur
- en fonction de la taille de l'entrée

Complexité d'un algorithme

Définition :

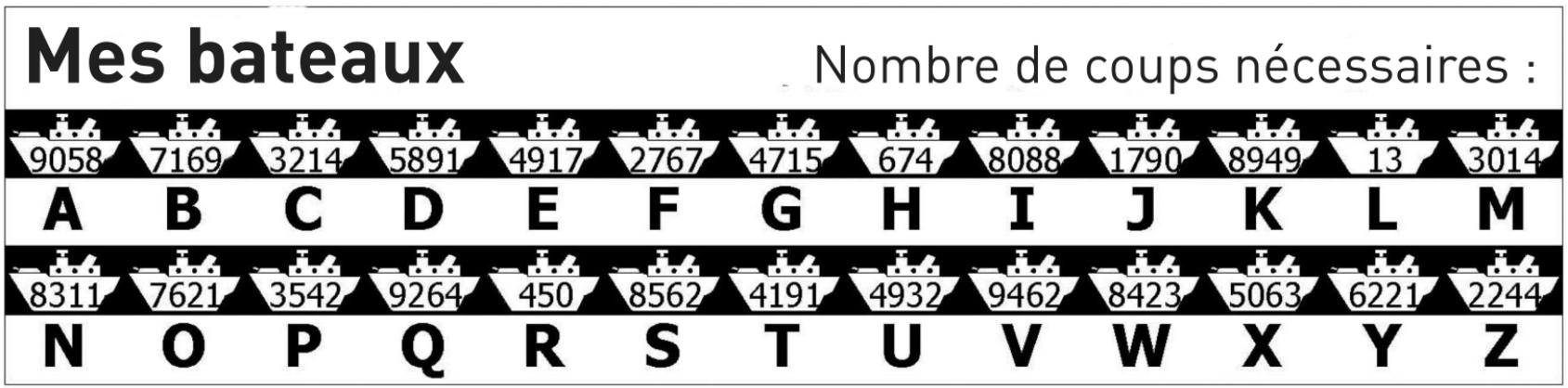
Complexité en temps : Quantité de temps nécessaire à l'algorithme pour s'exécuter

Complexité en espace : Quantité de mémoire nécessaire à l'algorithme pour s'exécuter.

- fortement dépendant de la machine, des éléments utilisés
- plutôt en terme d'ordre de grandeur
- en fonction de la taille de l'entrée

Notation de Landlau : complexité en $O(f(n))$
$\longrightarrow$ il existe une constante C telle que la complexité $\leq C \times f(n)$.

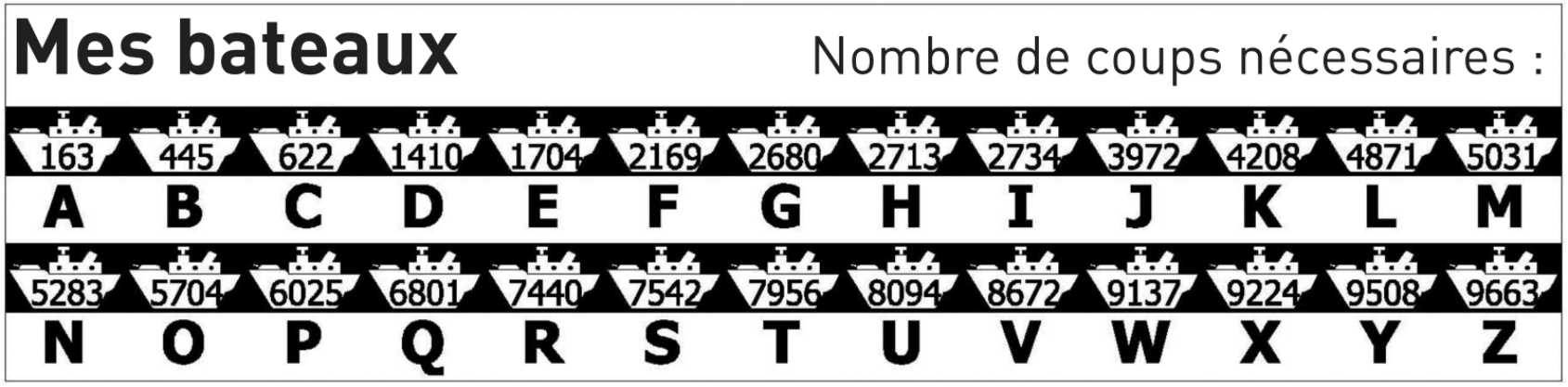
Retour sur la bataille navale :



Structure non organisée

- n appels dans le pire cas
- $\frac{n}{2}$ en moyenne
- Recherche en $O(n)$

Dichotomie

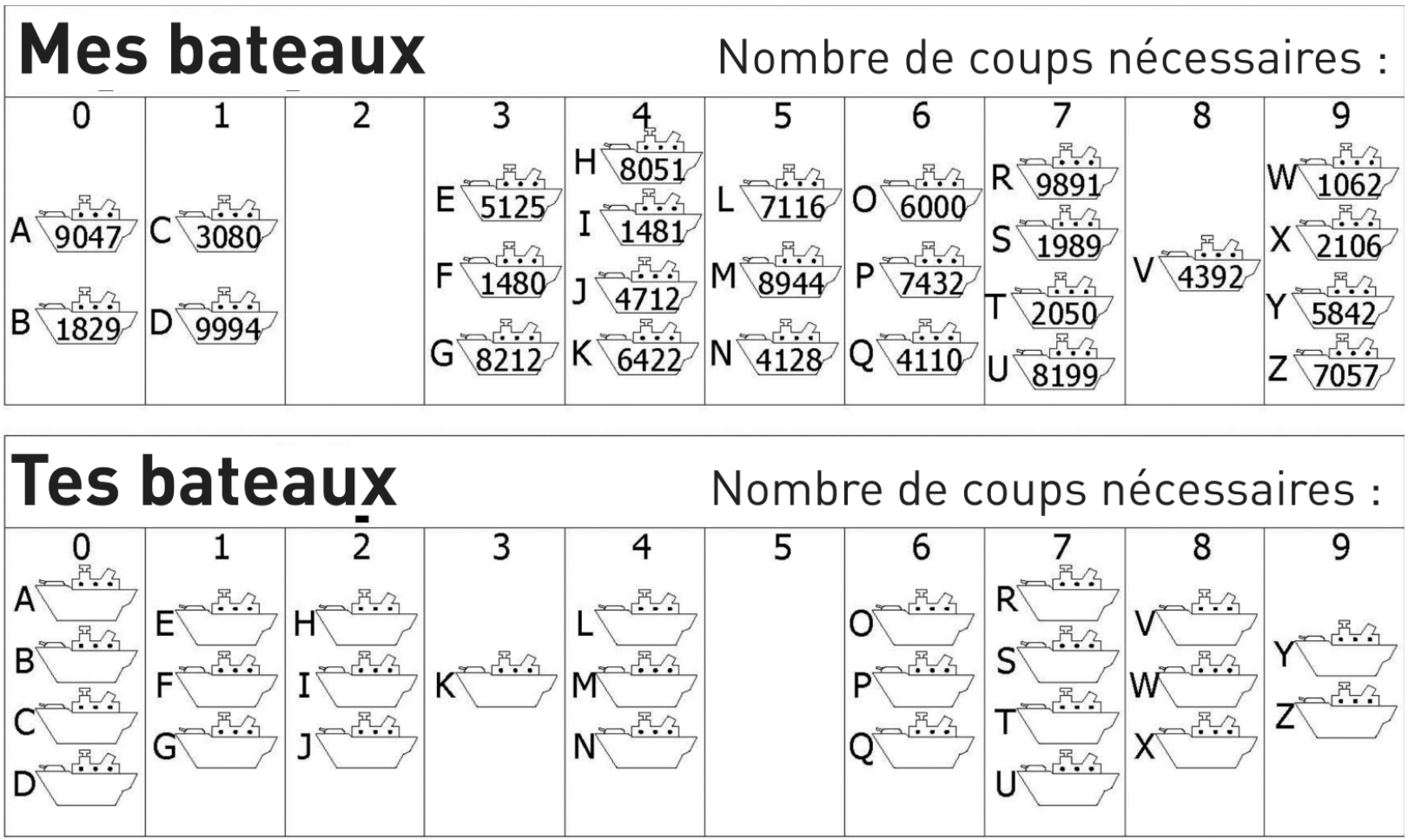


- Structure indexée.
- Éléments dans l'ordre.

Structure triée

- Recherche en $O(log_2(n))$ (dans le pire cas et en moyenne).

Encore une partie ?



Recherche par clé

- Pour chaque valeur, une clé est calculée (appelé *hash*).
- Pour chaque clé, l'ensemble des éléments qui correspondent (par exemple liste chainée).

Table de hashage

- Fort rôle de la taille de l'espace des clés disons k clés.
- Recherche en $O(\frac{n}{k})$ en moyenne.

Fonction de Hashage

Attributs recherchés

- clé décorélée de la valeur initiale (ensemble de valeurs homogènes)
- Peu de collisions (éléments ayant la même image)

Usages

- HashSet
- HashMap, dictionnaire (hash de la clé).
- Cryptographie (Signature, stockage de mot de passe)

Comparaison des structures de données

Tableau : pré-affecté en mémoire, n cases statiques (e.g. `ArrayList`)

Liste chaînée : structure chaînée, dynamique (e.g. `LinkedList`)

Table de hachage : Tableau de listes chaînées (e.g. `HashSet`)

Comparaison des structures de données

Complexité des opérations

Opération	Tableau	Liste	Tableau trié	Hashset
Accès par position Recherche Insertion/suppression				

Comparaison des structures de données

Complexité des opérations

Opération	Tableau	Liste	Tableau trié	Hashset
Accès par position	O(1)	O(n)	O(1)	N/A
Recherche	O(n)	O(n)	O(log(n))	O(n/k)
Insertion/suppression	O(n)	O(1)	O(n+log(n))	O(1)

En pratique, des techniques d'accélération.

Complexité en temps

Exemples et vocabulaire :

Problème	Complexité		$n = 50$
Accéder à une case d'un tableau	$O(1)$	[constant]	10ns
Chercher dans un tableau trié	$O(\log(n))$	[logarithmique]	20ns
Chercher dans un tableau	$O(n)$	[linéaire]	500ns
Multiplier des matrices	$O(n^3)$	[cubique]	1,25ms
Résoudre un Sudoku à n cases	$O(9^n)$	[exponentiel]	130 jours

Pour un problème dont l'entrée est de taille n

Algorithme polynomial : Complexité en $O(n^k)$, assez rapide

Algorithme Exponentiel : Compléxité en $O(k^n)$, très lent.

NP-completude

Culturel

Problème NP-complet

- Pas d'algorithme polynomial connu
- Si on en trouve un, on les résout tous.

Exemples de problèmes NP-complets

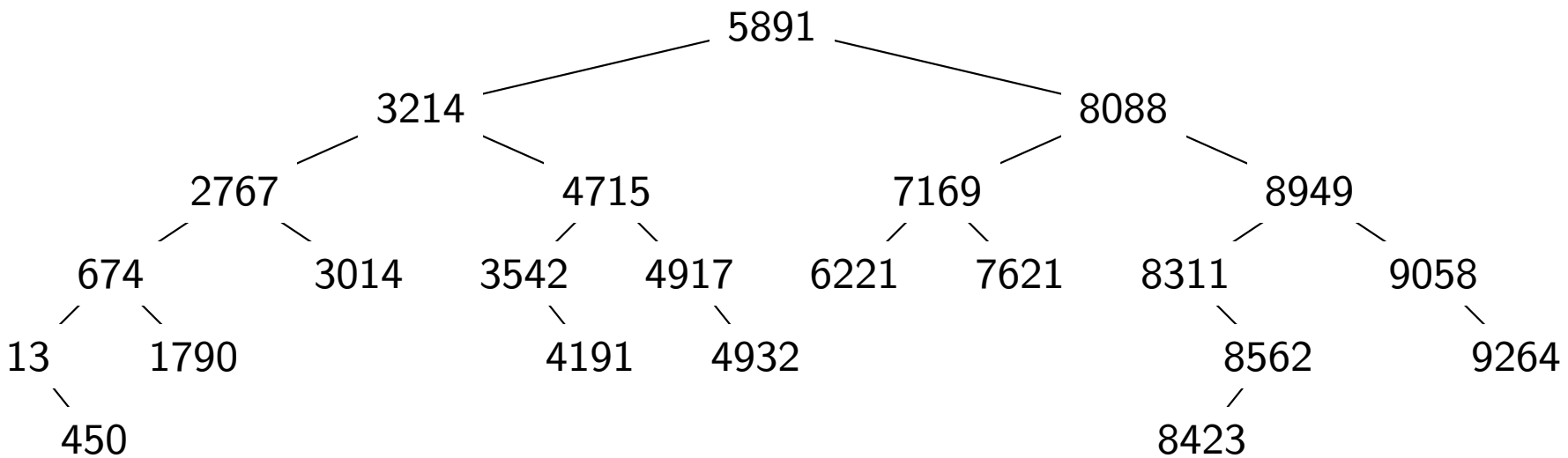
Voyageur de commerce plus court chemin qui visite n points fixés.

Emploi du temps qui satisfait les contraintes de salles/prof/élèves

Factoriser un grand nombre (permet de casser le chiffrement RSA)

Plein de casse-têtes (Sokoban, démineur, mario...)

Arbre binaire de recherche



Arbre binaire de recherche

Principes

- Structure d'arbre

Usages

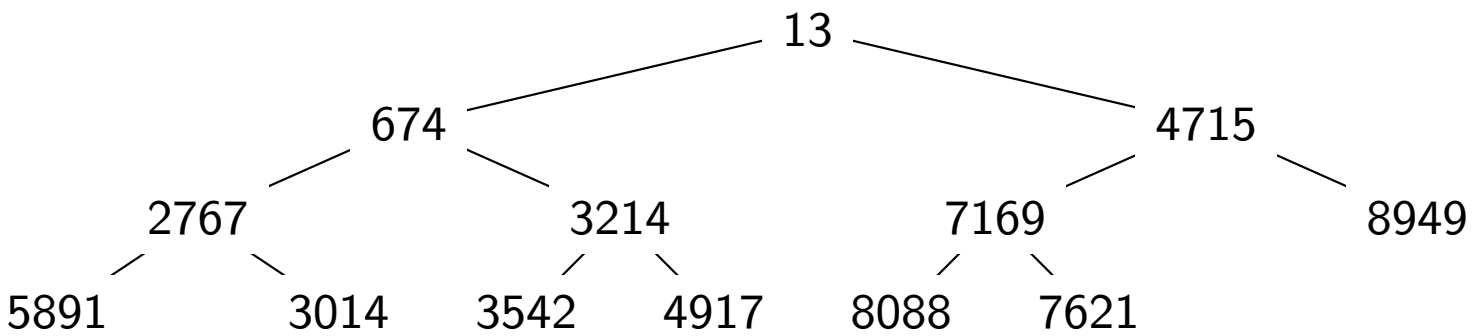
- insertion, recherche au même prix
- Complexité en moyenne : $O(\log(n))$
- Dans le pire cas : $O(n)$.
- Suppression délicate avec rotation

Extension de la rotation pour équilibrer l'arbre.

Les tas

Définition :

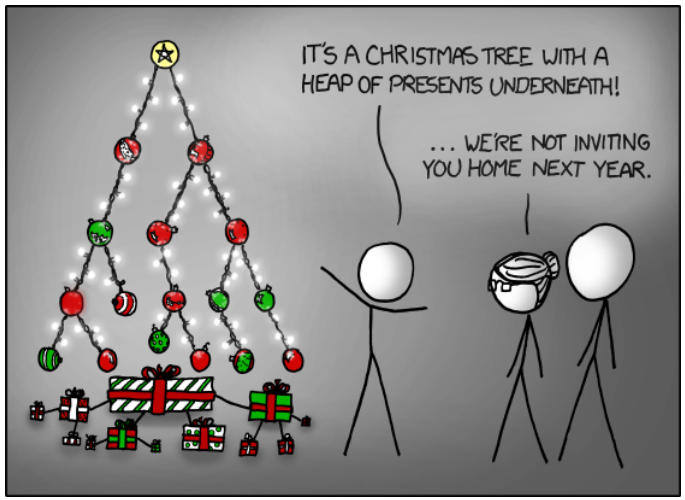
- Une structure semi-triée rangée dans un tableau
- Règle du jeu : chaque nœud est plus petit que ses enfants.
- en Java, implémentée dans les `PriorityQueue`.



Synthèse

À retenir

- Expression de la complexité algorithmique.
- Chacune des structures de données classiques.
- Savoir faire un choix éclairé !
- D'autres structures arborescentes existent, plus riches.



https://xkcd.com/835/