



LSINF1121 ALGORITHMIQUE ET STRUCTURES DE DONNÉES

TP1: COMPLEXITÉS, PILES, FILES ET LISTES

QUESTION 1.1.1: QU'EST-CE QU'UN TYPE ABSTRAIT DE DONNÉES?

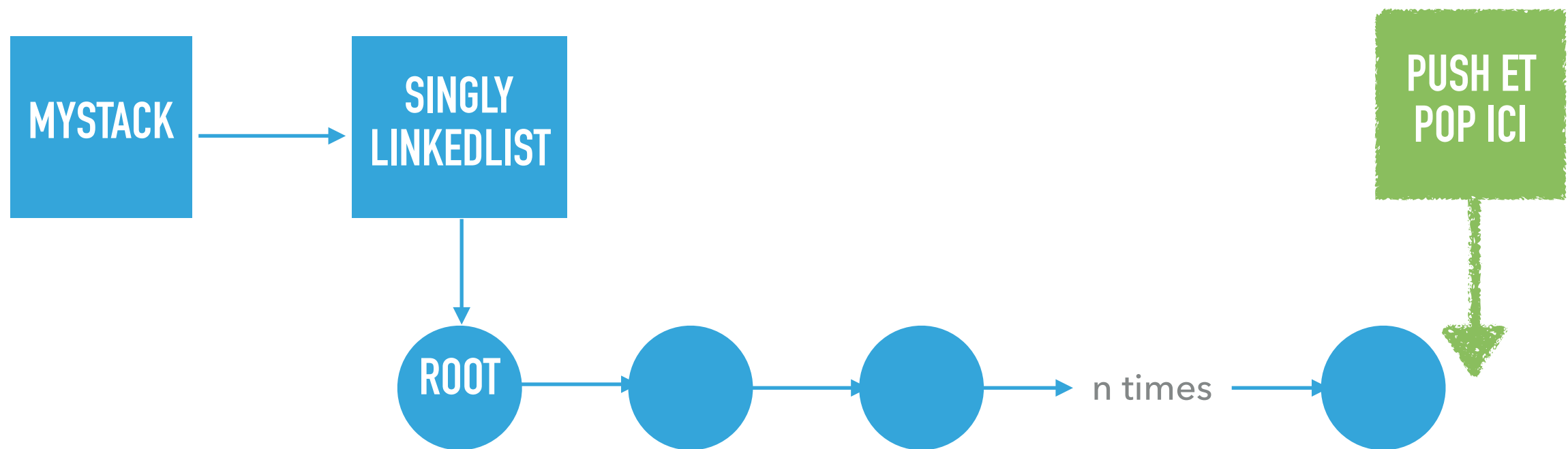
En informatique, un **type abstrait** est une spécification mathématique d'un ensemble de **données** et de l'ensemble des opérations qu'on peut effectuer sur elles. On qualifie d'**abstrait** ce **type** de **données** car il correspond à un cahier des charges qu'une structure de **données** doit ensuite implémenter.

DE QUOI SE RAPPROCHE LE PLUS UN TYPE ABSTRAIT DE DONNÉE EN JAVA?

- ▶ Une classe
- ▶ Un objet
- ▶ Une interface
- ▶ Une structure
- ▶ Un type primitif

QUESTION 1.1.2

Comment implémenter une pile (stack) avec une liste simplement chaînée, où les opérations se font en fin de liste?



QUELLE EST LA COMPLEXITÉ DE CETTE IMPLÉMENTATION?

$\mathcal{O}(1)$ ~ 1 $\Omega(1)$ $\Theta(1)$

$\mathcal{O}(n)$ $\sim n$ $\Omega(n)$ $\Theta(n)$

QUESTION 1.1.3: QUELLES SONT LES IMPLÉMENTATIONS POSSIBLES POUR UNE PILE?

Question ouverte, parce que je suis un fou.

QUESTION 1.1.3 (SUITE)

Pourquoi `java.util.Stack` est-il implémenté via un tableau?

| | Liste chainée | Tableau |
|--------|------------------|------------|
| Push | $O(1)$ | $O(n) ?$ |
| Pop | $O(1)$ | $O(n) ?$ |
| n push | $O(n)$ | $O(n^2) ?$ |
| n pop | $O(n)$ | $O(n^2) ?$ |

QUESTION 1.1.3 (SUITE)

Pourquoi java.util.Stack est-il implémenté via un tableau?

| | Liste chainée | Tableau |
|--------|------------------|---------------------------------|
| Push | $O(1)$ | $O(1)$ Sauf redim. $O(n)$ |
| Pop | $O(1)$ | $O(1)$ Sauf redim. $O(n)$ |
| n push | $O(n)$ | $O(n^2) ?$ |
| n pop | $O(n)$ | $O(n^2) ?$ |

On double la taille du tableau quand on atteint la limite.

Sur n push, on fait donc un resize à ces positions:

- ▶ 1
- ▶ 2
- ▶ 4
- ▶ 8
- ▶ 16
- ▶ ...
- ▶ n

Combien de redim.?

$$\sum_{i=0}^{\log_2 n} \frac{n}{2^i}$$

QUESTION 1.1.3 (SUITE)

Pourquoi java.util.Stack est-il implémenté via un tableau?

| | Liste chainée | Tableau |
|--------|------------------|---------------------------------|
| Push | $O(1)$ | $O(1)$ Sauf redim. $O(n)$ |
| Pop | $O(1)$ | $O(1)$ Sauf redim. $O(n)$ |
| n push | $O(n)$ | $O(n^2) ?$ |
| n pop | $O(n)$ | $O(n^2) ?$ |

On double la taille du tableau quand on atteint la limite.

Sur n push, on fait donc un resize à ces positions:

- ▶ 1
- ▶ 2
- ▶ 4
- ▶ 8
- ▶ 16
- ▶ ...
- ▶ n

Combien de redim.?

$$\sum_{i=0}^{\log_2 n} \frac{n}{2^i} < 2n \in \mathcal{O}(n)$$

QUESTION 1.1.3 (SUITE)

Pourquoi java.util.Stack est-il implémenté via un tableau?

| | Liste chainée | Tableau |
|--------|------------------|---------------------------------|
| Push | $O(1)$ | $O(1)$ Sauf redim. $O(n)$ |
| Pop | $O(1)$ | $O(1)$ Sauf redim. $O(n)$ |
| n push | $O(n)$ | $O(n)$ |
| n pop | $O(n)$ | $O(n)$ |

On double la taille du tableau quand on atteint la limite.

Sur n push, on fait donc un resize à ces positions:

- ▶ 1
- ▶ 2
- ▶ 4
- ▶ 8
- ▶ 16
- ▶ ...
- ▶ n

COMPLEXITÉS
AMORTIES

Combien de redim.?

$$\sum_{i=0}^{\log_2 n} \frac{n}{2^i} < 2n \in \mathcal{O}(n)$$

QUESTION 1.1.3 (SUITE)

Pourquoi `java.util.Stack` est-il implémenté via un tableau?

| | Liste chainée | Tableau |
|--------|------------------|---------------------------------|
| Push | $O(1)$ | $O(1)$ Sauf redim. $O(n)$ |
| Pop | $O(1)$ | $O(1)$ Sauf redim. $O(n)$ |
| n push | $O(n)$ | $O(n)$ (amorti) |
| n pop | $O(n)$ | $O(n)$ (amorti) |

Même complexités,
pourquoi ce choix
d'implémentation?

QUESTION 1.1.4: PILE AVEC DEUX FILES

Any ideas?

QUESTION 1.1.5: ITÉRATEURS

-> Code

QUESTION 1.1.6: TYPES DE COMPLEXITÉS

Le livre utilise la notation \sim (tilde). On utilise dans d'autres cours les notations O , Ω et Θ .

Quelle est la différence entre ces différentes classes de complexités?

QUESTION 1.1.6: TYPES DE COMPLEXITÉS

$$f(n) \in \mathcal{O}(g(n)) \iff \exists k \in \mathbb{R}^+, n_0 \in \mathbb{N} \text{ t.q. } f(n) \leq k \cdot g(n) \quad \forall n \geq n_0$$

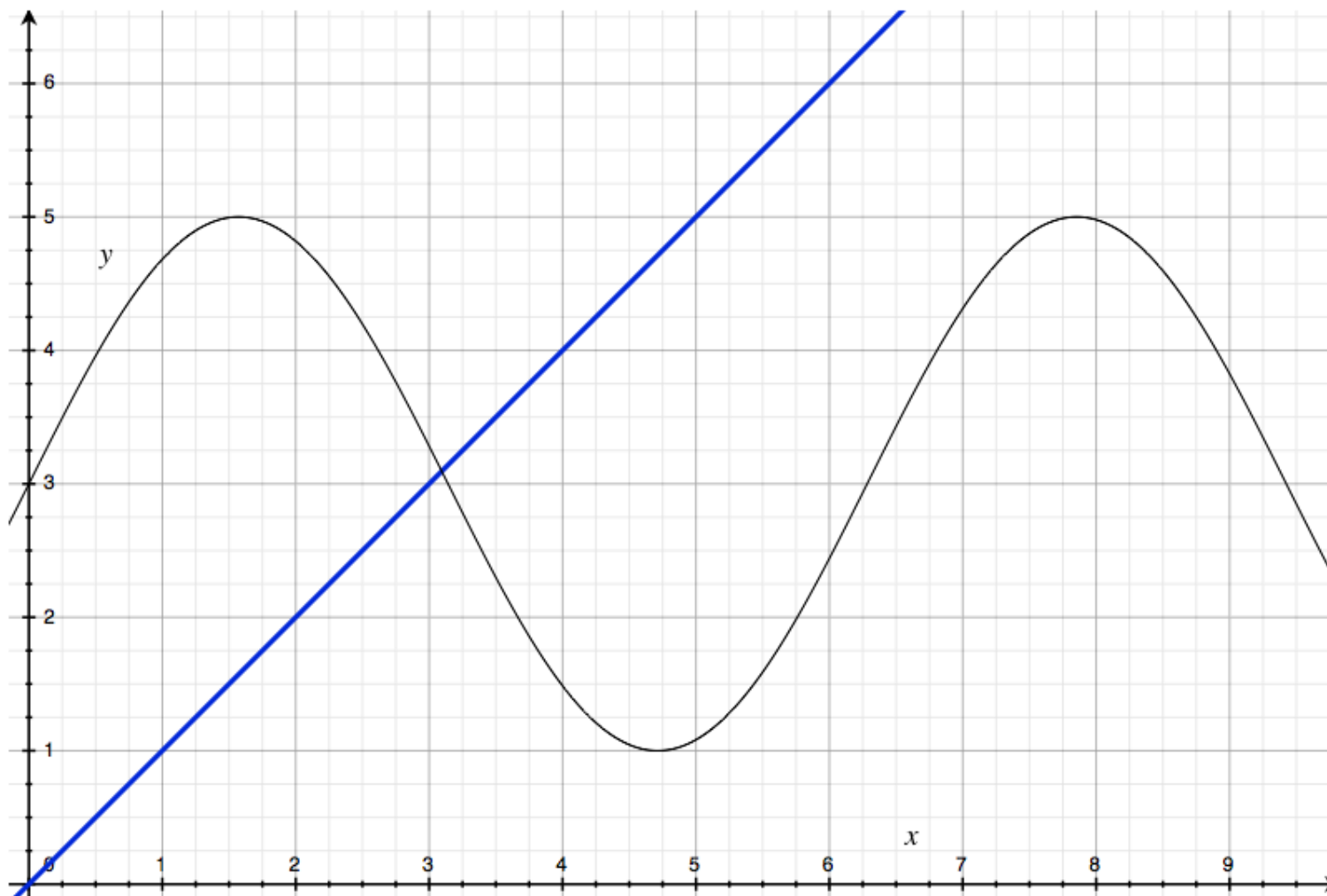
« Appartient »
 $\mathcal{O}()$ est donc un
ensemble

Ceci est une fonction
qui compte le nombre
d'opérations en
fonction de n

Il existe un k tel que
 $k \cdot g(n)$ est plus grand
que $f(n)$ quand n est
grand

QUESTION 1.1.6: TYPES DE COMPLEXITÉS

$$f(n) \in \mathcal{O}(g(n)) \iff \exists k \in \mathbb{R}^+, n_0 \in \mathbb{N} \text{ t.q. } f(n) \leq k \cdot g(n) \quad \forall n \geq n_0$$



$$f(x) = 2 \sin(x) + 2$$

$$g(x) = x$$

$$2 \sin(x) + 2 \in \mathcal{O}(x)$$

QUESTION 1.1.6: TYPES DE COMPLEXITÉS

$$f(n) \in \mathcal{O}(g(n)) \iff \exists k \in \mathbb{R}^+, n_0 \in \mathbb{N} \quad \text{t.q.} \quad f(n) \leq k \cdot g(n) \quad \forall n \geq n_0$$

$$f(n) \in \mathbf{\Omega}(g(n)) \iff \exists k \in \mathbb{R}^+, n_0 \in \mathbb{N} \quad \text{t.q.} \quad \mathbf{k} \cdot \mathbf{f(n)} \geq \mathbf{g(n)} \quad \forall n \geq n_0$$

$$f(n) \in \mathbf{\Theta}(g(n)) \iff \exists k_0, k_1 \in \mathbb{R}^+, n_0 \in \mathbb{N} \quad \text{t.q.} \quad \mathbf{k_0} \cdot \mathbf{g(n)} \leq \mathbf{f(n)} \leq \mathbf{k_1} \cdot \mathbf{g(n)} \quad \forall n \geq n_0$$

QUESTION 1.1.6: TYPES DE COMPLEXITÉS

$$f(n) \in \mathcal{O}(g(n)) \iff \exists k \in \mathbb{R}^+, n_0 \in \mathbb{N} \quad \text{t.q.} \quad f(n) \leq k \cdot g(n) \quad \forall n \geq n_0$$

$$f(n) \in \mathbf{\Omega}(g(n)) \iff \exists k \in \mathbb{R}^+, n_0 \in \mathbb{N} \quad \text{t.q.} \quad \mathbf{k} \cdot \mathbf{f(n)} \geq \mathbf{g(n)} \quad \forall n \geq n_0$$

$$f(n) \in \mathbf{\Theta}(g(n)) \iff \exists k_0, k_1 \in \mathbb{R}^+, n_0 \in \mathbb{N} \quad \text{t.q.} \quad \mathbf{k_0} \cdot \mathbf{g(n)} \leq \mathbf{f(n)} \leq \mathbf{k_1} \cdot \mathbf{g(n)} \quad \forall n \geq n_0$$

$$f(n) \sim g(n) \iff \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1$$

PLEINS DE QUESTIONS SUR WOOCCLAP

Courage, c'est ici que ça se complique.

wooclap

QUESTION 1.1.7: DOUBLING RATIO TEST

$$\text{Si } f(n) \sim an^b \log n \quad \text{alors} \quad \frac{f(2n)}{f(n)} \sim 2^b$$

QUESTION 1.1.7: DOUBLING RATIO TEST

| n | 1000 | 2000 | 4000 | 8000 | 16000 | 32000 | 64000 |
|-----------------|------|------|------|------|-------|-------|-------|
| T(n) | 0 | 0 | 0.1 | 0.3 | 1.3 | 5.1 | 20.5 |
| Ratio précédent | | ~1 | ~1 | ~3 | 4.3 | 3.92 | 4.01 |