

# Systemes de fichiers

Thomas Lavergne  
lavergne@lisn.fr

## Système de fichiers

- Découpage des fichiers en **blocs logiques**
- **Allocation** des blocs sur le support **physique**

## Système de fichiers

- Découpage des fichiers en **blocs logiques**
- **Allocation** des blocs sur le support **physique**

## Problèmes

- Accès aux blocs
  - Minimiser le temps de réponse du matériel
  - Dépend du matériel et des algorithmes
- Garantir l'intégrité des données
  - Vérifier les blocs

## Principe

Le disque est beaucoup plus lent que la RAM

- Ne pas bloquer le processeur pendant le chargement des blocs

→ tampon et transfert par bloc

## Principe

Le disque est beaucoup plus lent que la RAM

- Ne pas bloquer le processeur pendant le chargement des blocs

→ tampon et transfert par bloc

## Tampon

Espace de stockage plus petit mais plus rapide conservant les données les plus utilisées

## Principe

Le disque est beaucoup plus lent que la RAM

- Ne pas bloquer le processeur pendant le chargement des blocs
- tampon et transfert par bloc

## Tampon

Espace de stockage plus petit mais plus rapide conservant les données les plus utilisées

## Utilisation

- Gestion : cf. algos remplacement de pages
- Utilisé pour pages et blocs disques

## Détection des secteurs défectueux

Secteur défectueux  $\rightarrow$  relecture  $\neq$  écriture

## Détection des secteurs défectueux

Secteur défectueux  $\rightarrow$  relecture  $\neq$  écriture

## Code d'erreur

**Fonction** de l'ensemble des données du bloc

- Stocké sur le secteur
- Comparé avec  $\phi(\text{données\_secteur})$

## Détection des secteurs défectueux

Secteur défectueux  $\rightarrow$  relecture  $\neq$  écriture

## Code d'erreur

**Fonction** de l'ensemble des données du bloc

- Stocké sur le secteur
- Comparé avec  $\phi(\text{données\_secteur})$

## Exemple: Somme de contrôle

2 bits de données  $\rightarrow$  1 bit de **parité de la somme**

- 0001 1011 0110 1100  $\rightarrow^{\phi}$  0110 1100
- Vérification: 000  $\rightarrow$  ok, 010  $\rightarrow$  **erreur**

## Détection des secteurs défectueux

Secteur défectueux  $\rightarrow$  relecture  $\neq$  écriture

## Code d'erreur

**Fonction** de l'ensemble des données du bloc

- Stocké sur le secteur
- Comparé avec  $\phi(\text{données\_secteur})$

## Exemple: Somme de contrôle

2 bits de données  $\rightarrow$  1 bit de **parité de la somme**

- 0001 1011 0110 1100  $\rightarrow^{\phi}$  0110 1100
- Vérification: 000  $\rightarrow$  ok, 010  $\rightarrow$  **erreur**

*mais on ne peut pas savoir lequel des 3 bits a été modifié. . .*

Disques durs

Fiches perforées (années 50)

X Capacité (nombre de trous)

### Fiches perforées (années 50)

X Capacité (nombre de trous)

### Bandes magnétiques (années 60)

✓ Capacité

X Allocation contiguë

## Fiches perforées (années 50)

X Capacité (nombre de trous)

## Bandes magnétiques (années 60)

✓ Capacité

X Allocation contiguë

## Disques durs/disquettes (années 70)

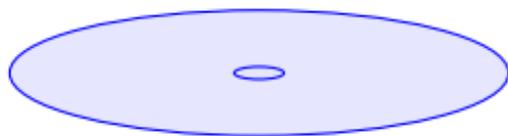
✓ Capacité

✓ Allocation libre

X Fragilité

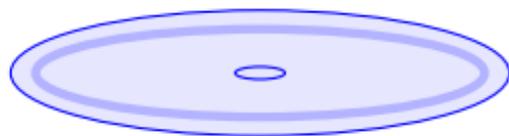
## Disque

- Plaque circulaire



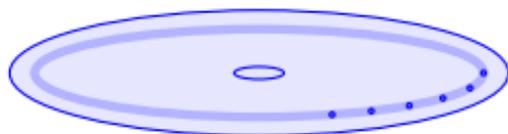
## Disque

- Plaque circulaire  $\rightarrow$  n **pistes** concentriques



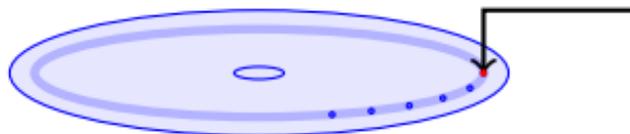
## Disque

- Plaque circulaire  $\rightarrow$   $n$  **pistes** concentriques
- $m$  **secteurs** (cadre de blocs) par piste



## Disque

- Plaque circulaire  $\rightarrow$   $n$  **pistes** concentriques
- $m$  **secteurs** (cadre de blocs) par piste
- Tête de lecture **mobile**

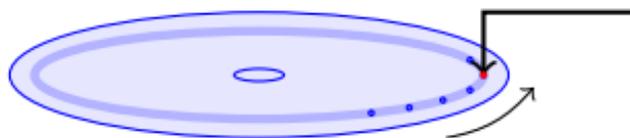


## Fonctionnement

- La tête lit **1 secteur à la fois**

## Disque

- Plaque circulaire  $\rightarrow$   $n$  **pistes** concentriques
- $m$  **secteurs** (cadre de blocs) par piste
- Tête de lecture **mobile**

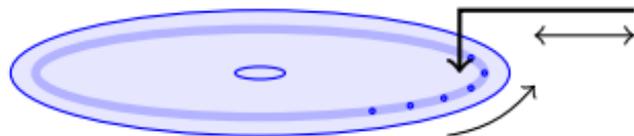


## Fonctionnement

- La tête lit **1 secteur à la fois**
- Rotation disque  $\rightarrow$  lecture des secteurs de la **piste**

## Disque

- Plaque circulaire  $\rightarrow$   $n$  **pistes** concentriques
- $m$  **secteurs** (cadre de blocs) par piste
- Tête de lecture **mobile**



## Fonctionnement

- La tête lit **1 secteur à la fois**
- Rotation disque  $\rightarrow$  lecture des secteurs de la **piste**
- Déplacement tête  $\rightarrow$  lecture des **autres pistes**

### Nombre de secteurs par piste

Les pistes concentriques sont toutes de tailles différentes !

- Combien de secteurs par piste ?

### Nombre de secteurs par piste

Les pistes concentriques sont toutes de tailles différentes !

- Combien de secteurs par piste ?

### Nombre de secteurs variable ?

- Difficile à gérer pour le contrôleur !

Ex : conversion bloc logique  $\rightarrow$  adresse physique (piste, secteur)

- Temps d'accès variable selon piste !

### Nombre de secteurs par piste

Les pistes concentriques sont toutes de tailles différentes !

- Combien de secteurs par piste ?

### Nombre de secteurs variable ?

- Difficile à gérer pour le contrôleur !

Ex : conversion bloc logique  $\rightarrow$  adresse physique (piste, secteur)

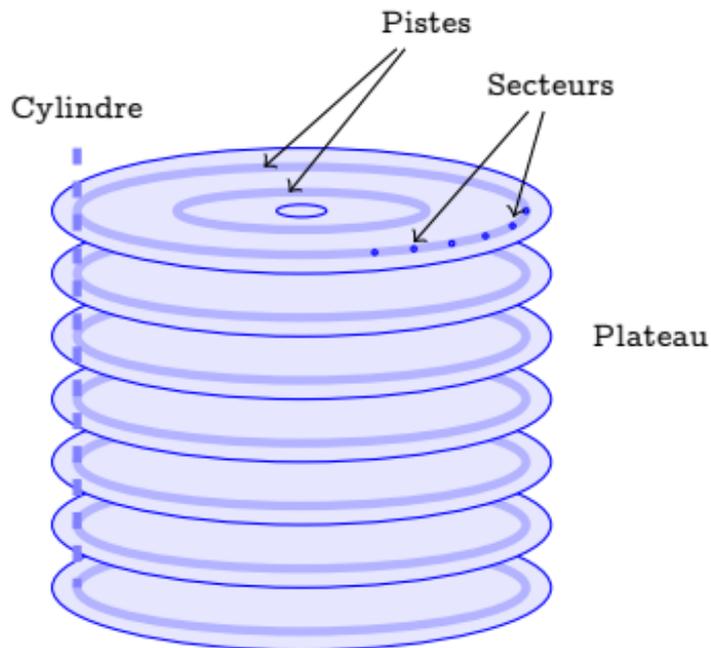
- Temps d'accès variable selon piste !

**Le nombre de secteurs par pistes est constant**

(ils sont plus ou moins étalés)

## Cylindres

Plusieurs disques empilés, appelés **plateaux**  
Les pistes de même rayon forment un **cylindre**



### Disque dur

- Un disque dur est composé de  $n$  cylindres
- Chaque cylindre est composé de  $m$  pistes
- Chaque piste est composée de  $k$  secteurs

## Disque dur

- Un disque dur est composé de  $n$  cylindres
- Chaque cylindre est composé de  $m$  pistes
- Chaque piste est composée de  $k$  secteurs

## Tête de lecture

La tête de lecture est composée de :

- Un bras mobile en **rateau**
  - $m$  têtes **fixes** aux extrémités
  - Un **multiplexeur** pour sélectionner la piste à lire
- Chaque tête s'insère au dessus d'une piste

### Secteurs

Un **secteur** peut contenir un seul **bloc** de données

## Secteurs

Un **secteur** peut contenir un seul **bloc** de données

## Numérotation

- Le bloc 0 est sur le premier secteur de la première piste du cylindre supérieur
- Par secteur croissant, puis par piste croissante, puis par cylindre

## Secteurs

Un **secteur** peut contenir un seul **bloc** de données

## Numérotation

- Le bloc 0 est sur le premier secteur de la première piste du cylindre supérieur
- Par secteur croissant, puis par piste croissante, puis par cylindre

## Bloc logique

- Adresse physique = (**cylindre,piste,secteur**)
- Bloc logique  $\rightarrow$  adresse physique

### Secteur défectueux

Dans tout support physique, certains secteurs deviennent inutilisables avec le temps

### Secteur défectueux

Dans tout support physique, certains secteurs deviennent inutilisables avec le temps

### Table des blocs

- Marquer le secteur inutilisable
- Modifier l'adresse physique du bloc logique  
→ Il faut une table des blocs

### Secteur défectueux

Dans tout support physique, certains secteurs deviennent inutilisables avec le temps

### Table des blocs

- Marquer le secteur inutilisable
- Modifier l'adresse physique du bloc logique  
→ Il faut une table des blocs

### Méthodes de gestion

- par l'OS + marquer secteurs défectueux dans la FAT
- par le contrôleur du périphérique

### Structures de données

- Table des secteurs défectueux
- Réserver un **ensemble de secteurs pour des remplacement** lors du formatage du disque

### Structures de données

- Table des secteurs défectueux
- Réserver un **ensemble de secteurs pour des remplacement** lors du formatage du disque

### Contrôleur de périphérique

- Vérification à l'écriture
- Préviend l'OS d'un secteur défectueux
- L'OS demande un remplacement (**glissement**)
- Accès transparent pour l'OS qui ne voit que des blocs **logiques**

Ordonnancement

## Accès à un secteur

- Positionnement de la tête de lecture sur la piste  
*donc sur le bon cylindre...*
- Rotation du disque (maximum un tour d'attente)

## Accès à un secteur

- Positionnement de la tête de lecture sur la piste  
*donc sur le bon cylindre...*
- Rotation du disque (maximum un tour d'attente)

## Temps d'accès

- Le disque tourne en permanence
- Vitesse de rotation = caractéristique matérielle
- Action sur les déplacements entre cylindres

## Accès à un secteur

- Positionnement de la tête de lecture sur la piste  
*donc sur le bon cylindre...*
- Rotation du disque (maximum un tour d'attente)

## Temps d'accès

- Le disque tourne en permanence
- Vitesse de rotation = caractéristique matérielle
- Action sur les déplacements entre cylindres

## Principe

Minimiser le temps de déplacement de la tête en parcourant les cylindres dans un ordre intelligent

### Caractéristique disque

- Vitesse de rotation fixée
- 256 cylindres

### Caractéristique disque

- Vitesse de rotation fixée
- 256 cylindres

### Requêtes

Au temps 0, la tête est sur le cylindre 53.

Le contrôleur reçoit les demandes suivantes :

98, 183, 37, 122, 14, 124, 65, 67

### Caractéristique disque

- Vitesse de rotation fixée
- 256 cylindres

### Requêtes

Au temps 0, la tête est sur le cylindre 53.

Le contrôleur reçoit les demandes suivantes :

98, 183, 37, 122, 14, 124, 65, 67

### Ordonnancement optimal

Pas d'autre demande → trier et partir du plus proche

En pratique : la file est **dynamique**

## Principe

Prendre les cylindres dans l'ordre

## Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

53

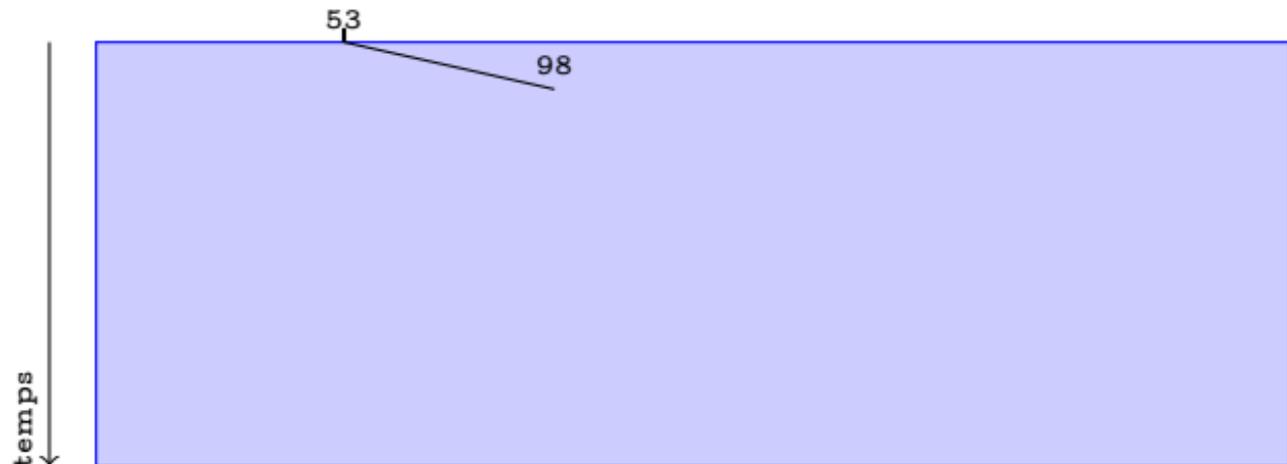


# First Come, First Served

## Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



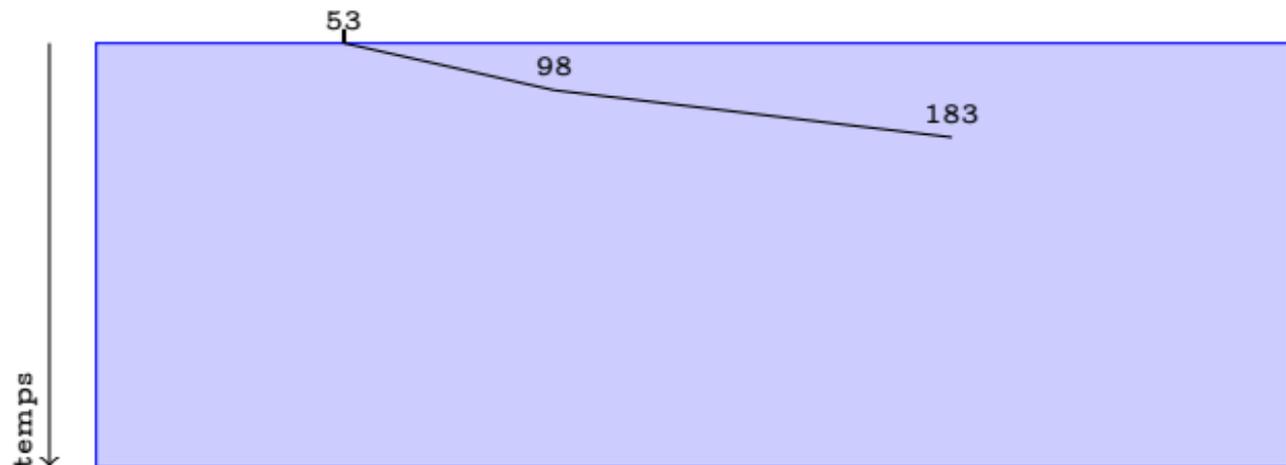
45+

# First Come, First Served

## Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



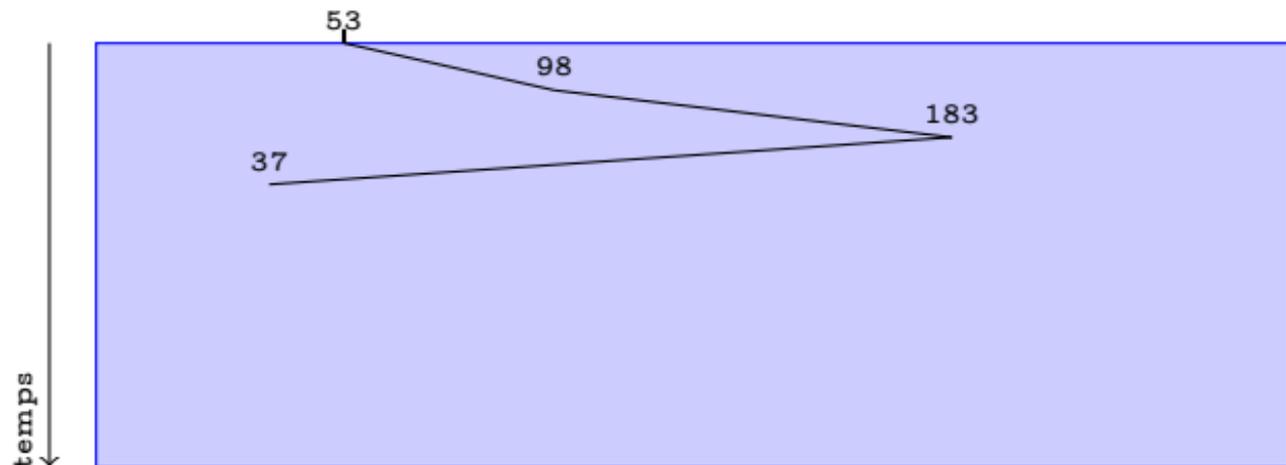
45+85+

# First Come, First Served

## Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

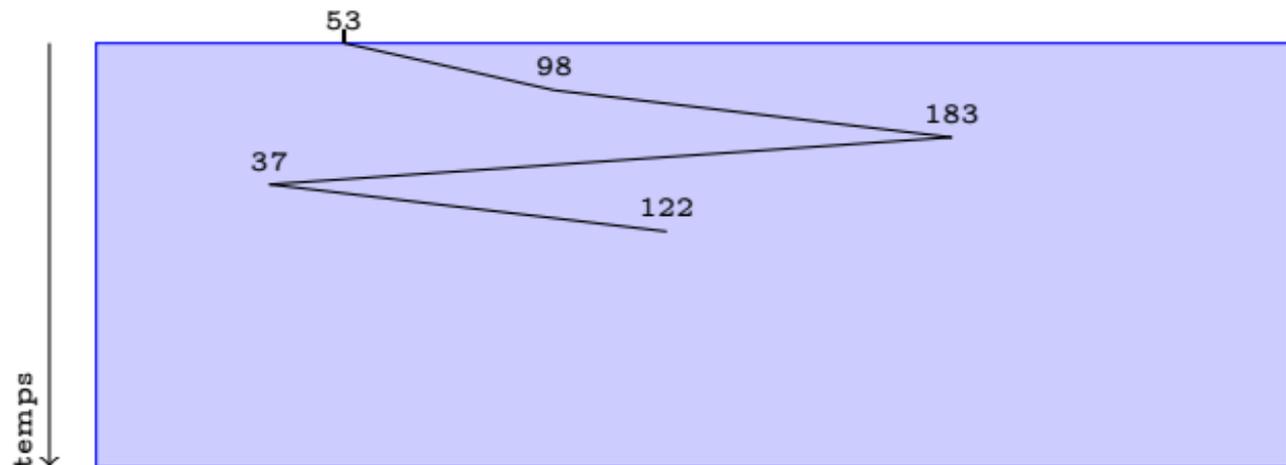


45+85+146+

## Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

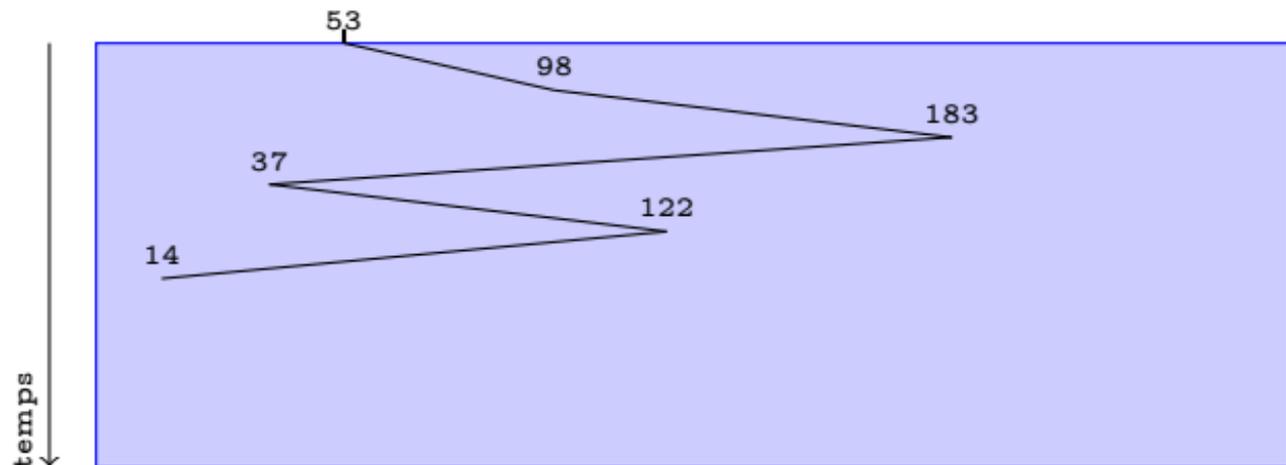


45+85+146+85+

## Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

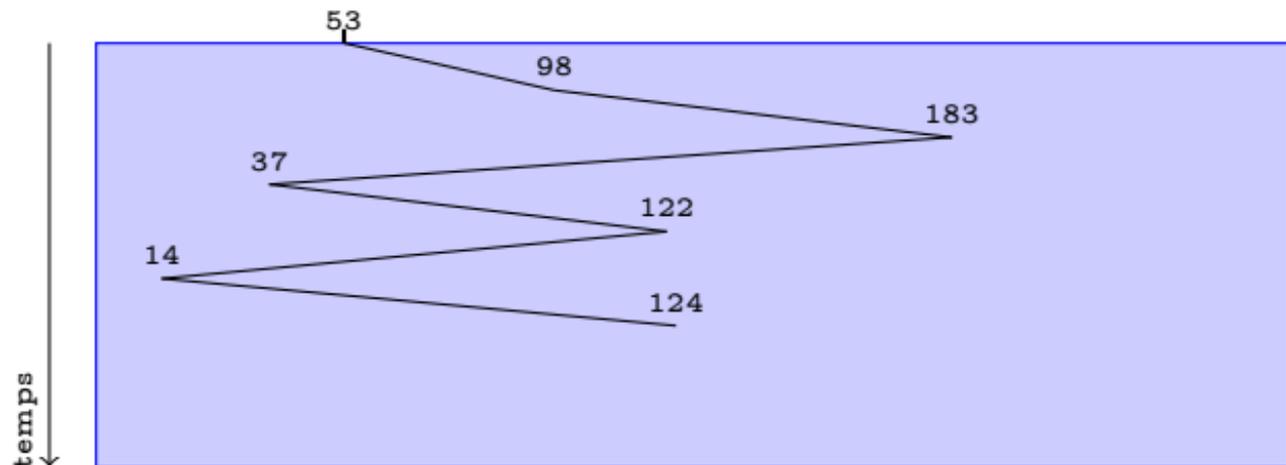


45+85+146+85+108+

## Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

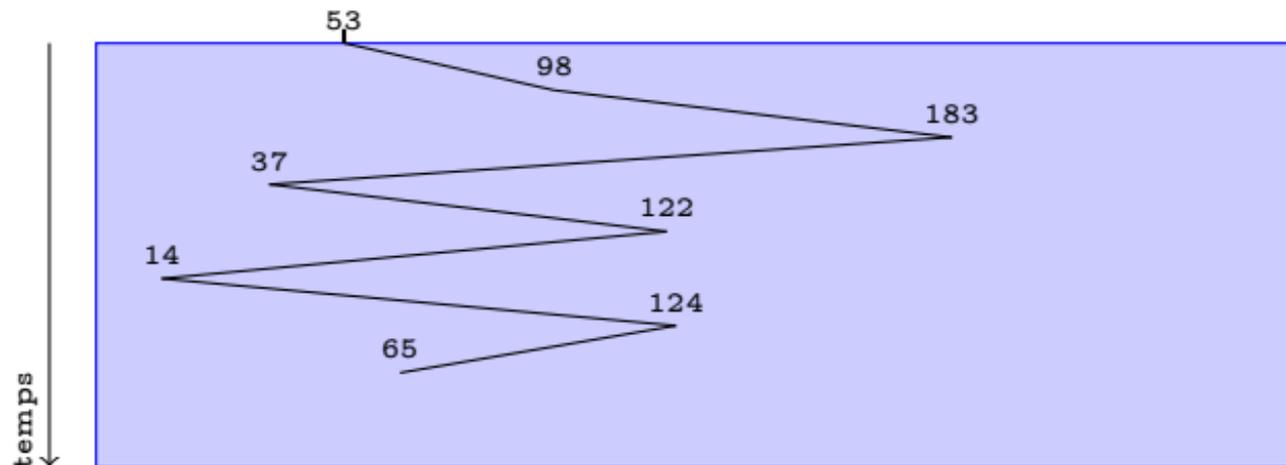


45+85+146+85+108+110+

## Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

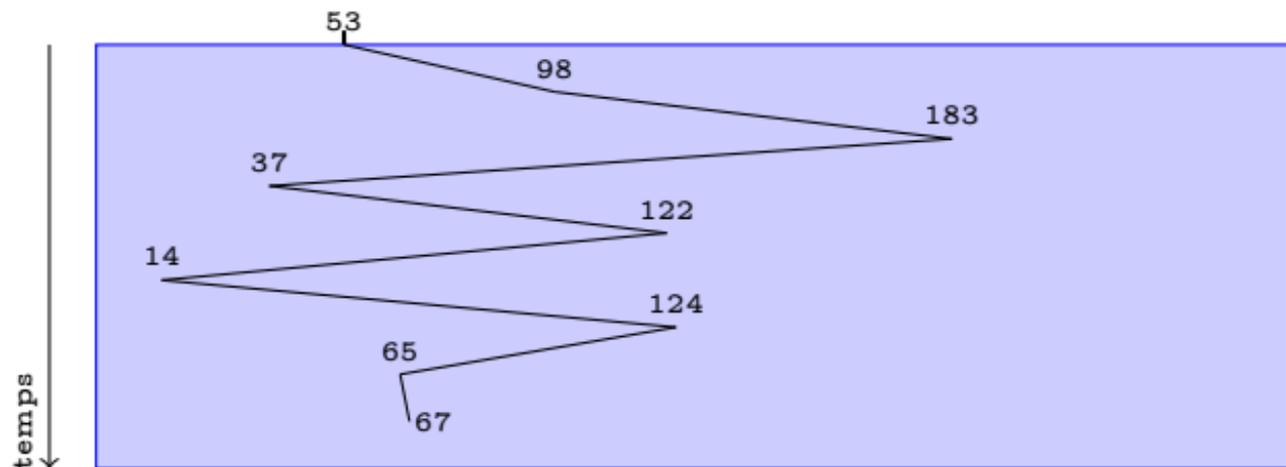


45+85+146+85+108+110+59+

## Principe

Prendre les cylindres dans l'ordre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



$45 + 85 + 146 + 85 + 108 + 110 + 59 + 2 = 640$  cylindres parcourus

## Principe

Aller vers le cylindre le plus proche

# Shortest Seek Time First

## Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

53



# Shortest Seek Time First

## Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



12+

# Shortest Seek Time First

## Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



12+2+

# Shortest Seek Time First

## Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



12+2+30+

# Shortest Seek Time First

## Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



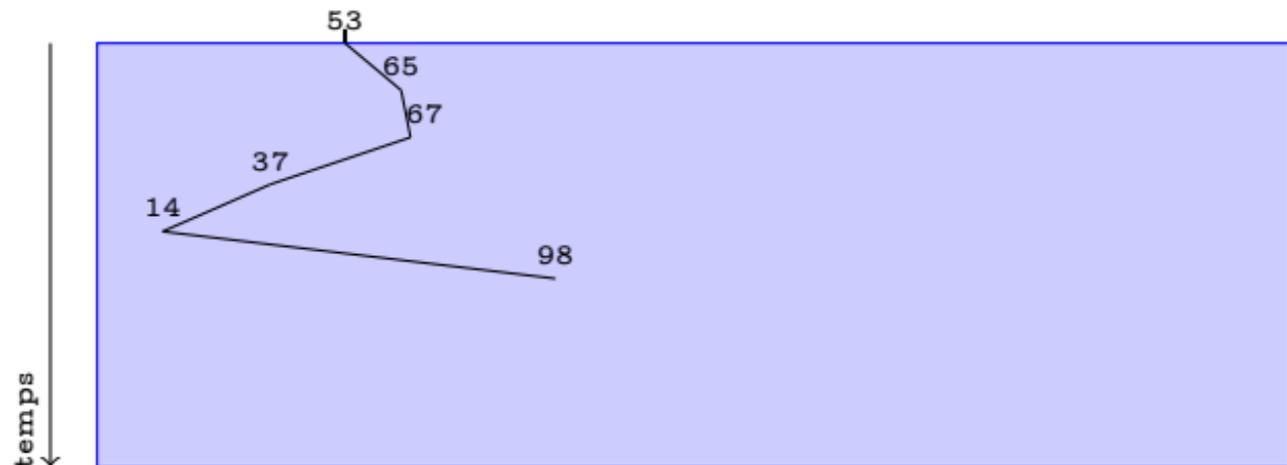
12+2+30+23+

# Shortest Seek Time First

## Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



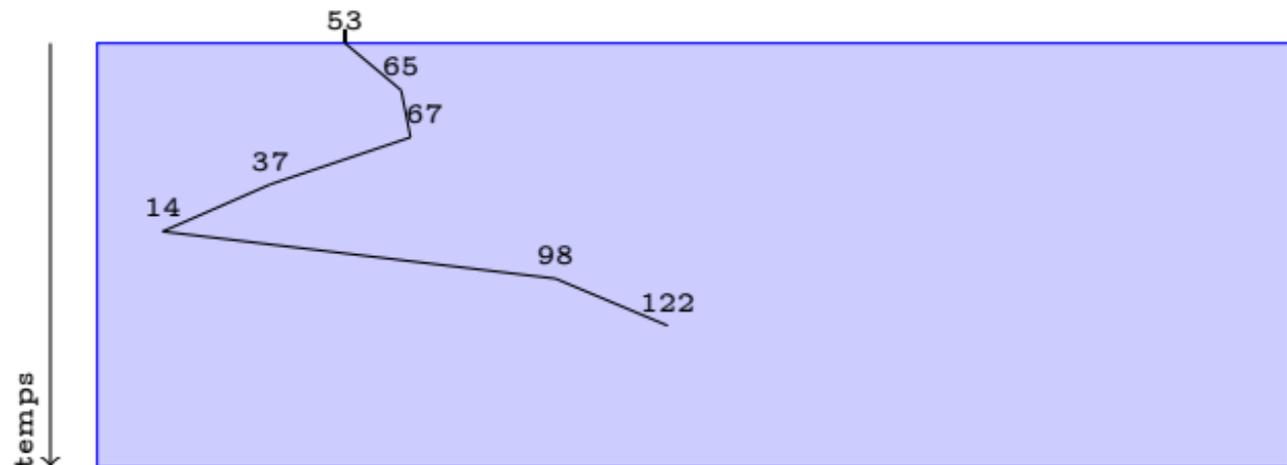
12+2+30+23+84+

# Shortest Seek Time First

## Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



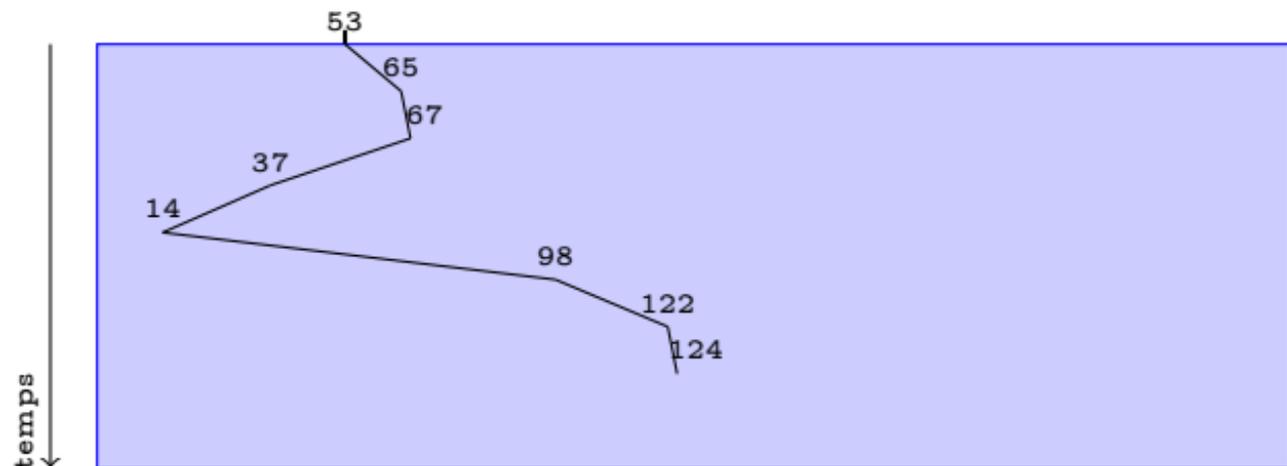
12+2+30+23+84+24+

# Shortest Seek Time First

## Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



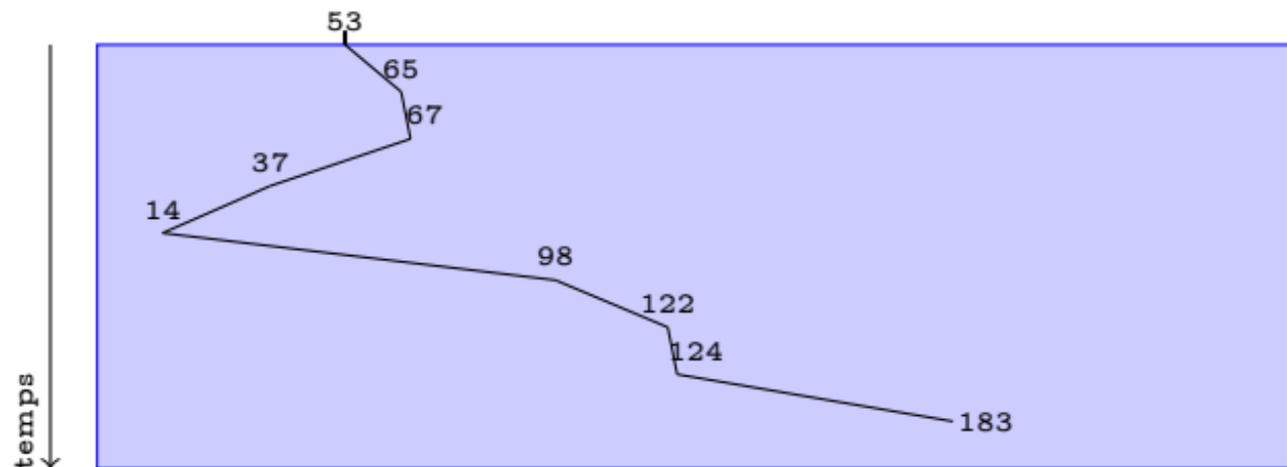
12+2+30+23+84+24+2+

# Shortest Seek Time First

## Principe

Aller vers le cylindre le plus proche

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



$12+2+30+23+84+24+2+59=236$  cylindres parcourus

## Avantages

- ✓ Temps de traitement souvent très bon

## Avantages

- ✓ Temps de traitement souvent très bon

## Limites

- ✗ Pas forcément optimal...
  - Il faut tenir compte des nouvelles arrivées

## Avantages

- ✓ Temps de traitement souvent très bon

## Limites

- X Pas forcément optimal...  
→ Il faut tenir compte des nouvelles arrivées
- X Risque de **famine**!

*Tant qu'il arrive des cylindres proches, on reste dans la zone et les autres cylindres ne sont pas servis!*

## Principe

Balayer dans un sens puis dans l'autre

## Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

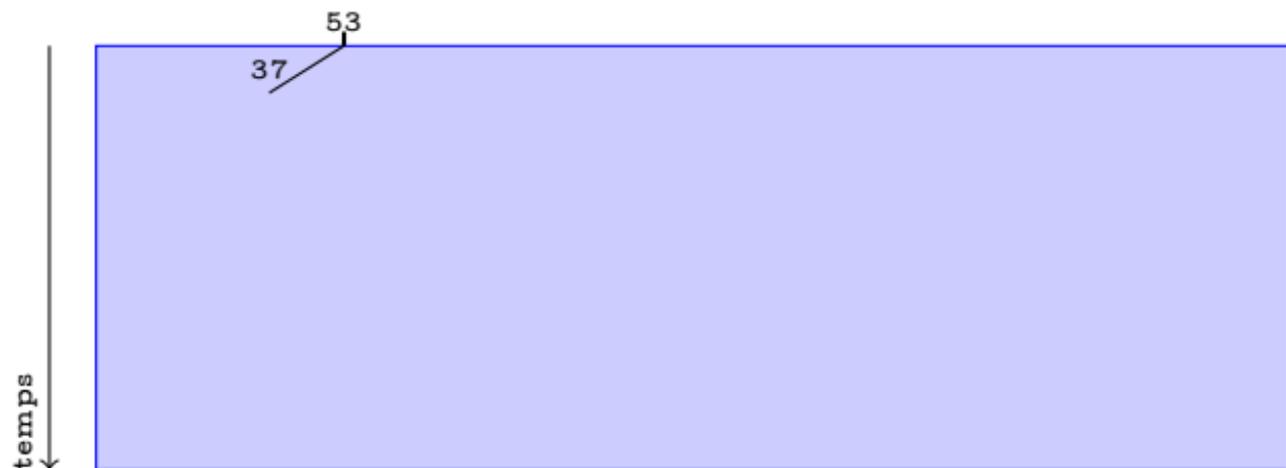
53



## Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

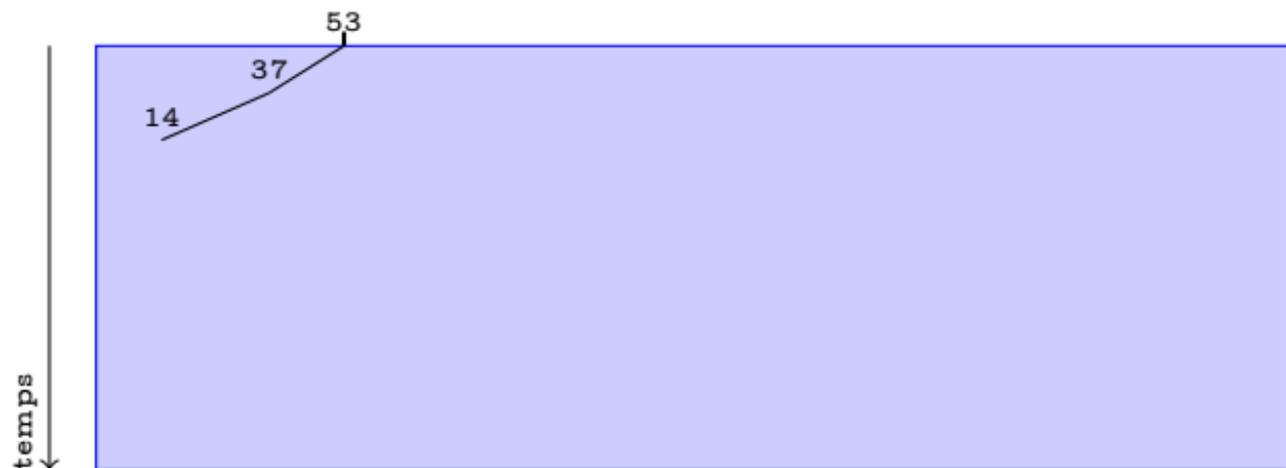


16+

## Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

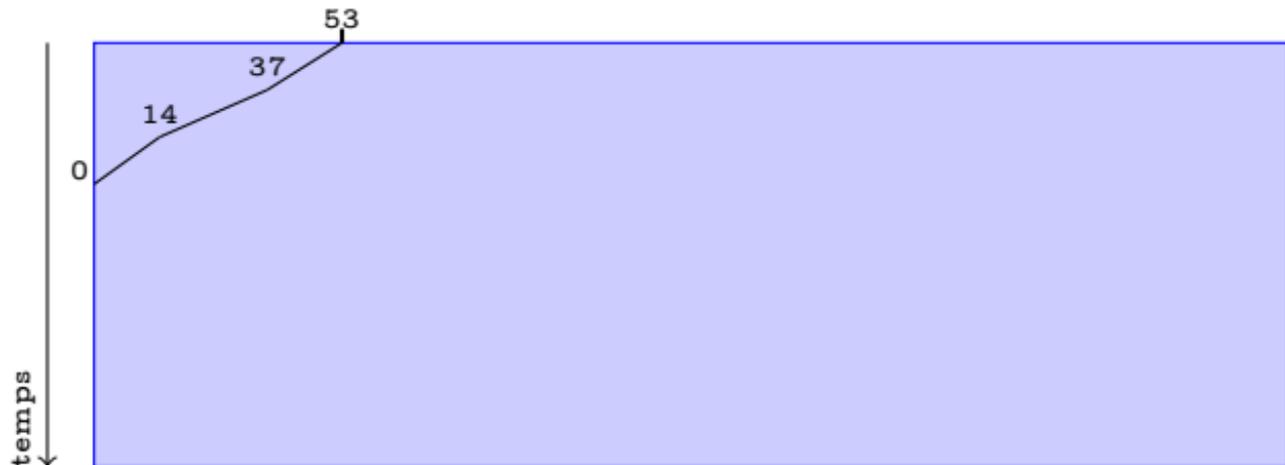


16+23+

## Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



16+23+14+

## Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



16+23+14+65+

## Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

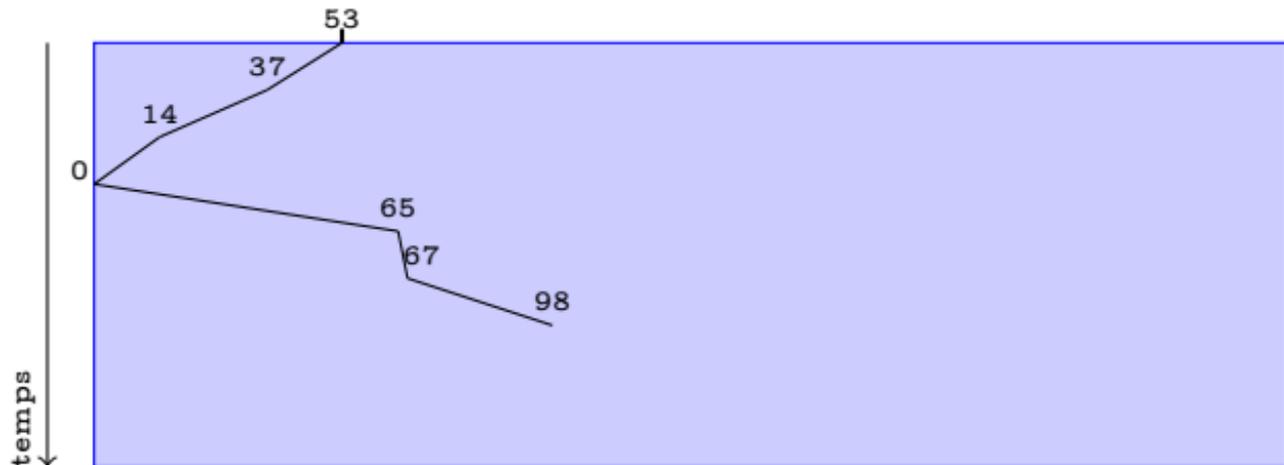


16+23+14+65+2+

## Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

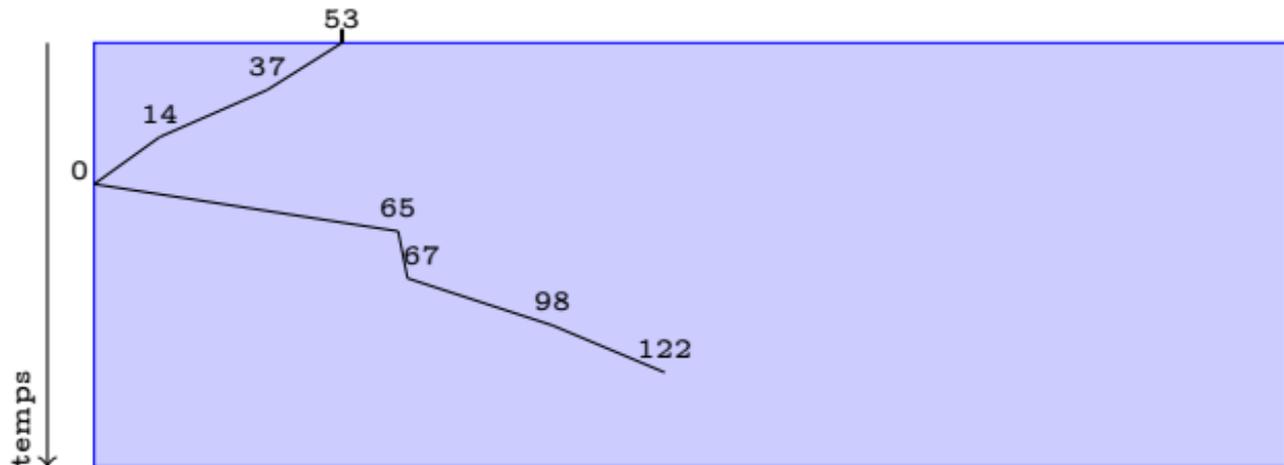


16+23+14+65+2+31+

## Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

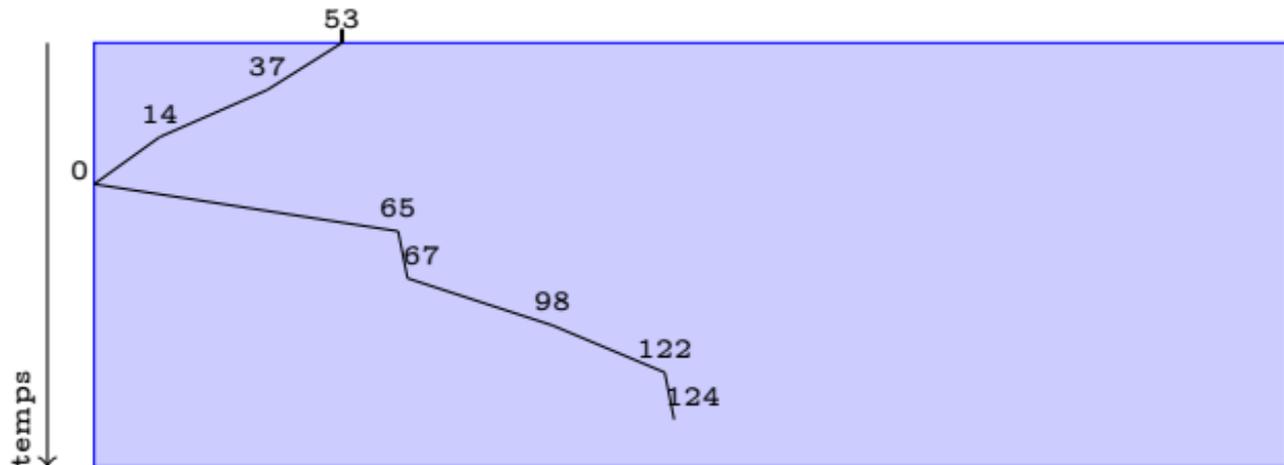


16+23+14+65+2+31+24+

## Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

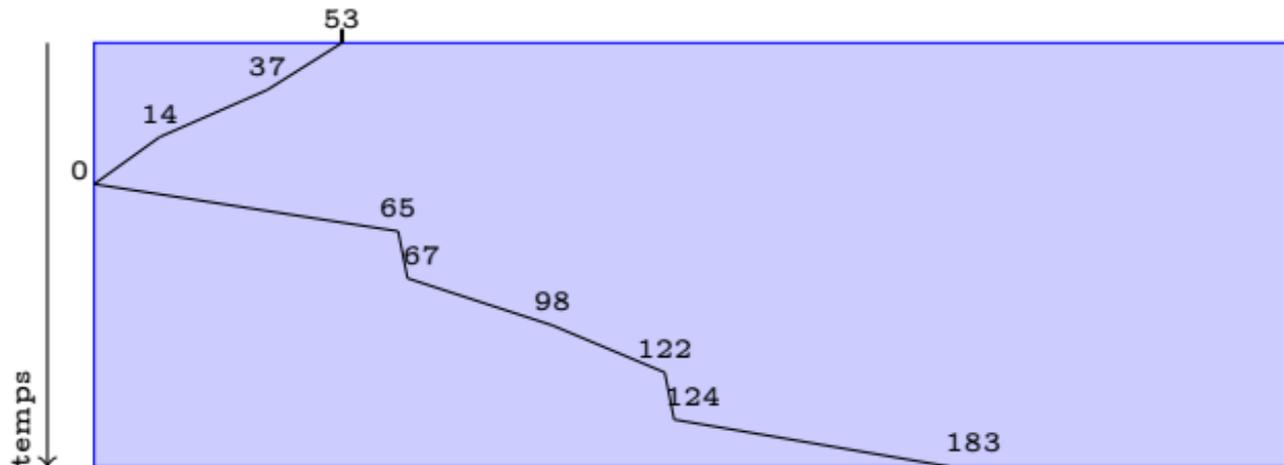


16+23+14+65+2+31+24+2+

## Principe

Balayer dans un sens puis dans l'autre

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



$$16+23+14+65+2+31+24+2+59 = 53 + 183 = 236$$

## Avantages

- ✓ Temps de traitement souvent très bon
- ✓ Pas de famine

## Avantages

- ✓ Temps de traitement souvent très bon
- ✓ Pas de famine

## Limites

- Parcours inutiles vers les bords
- Lorsqu'on fait demi-tour, on vient de servir les cylindres près du bord → il est peu probable d'en avoir beaucoup à traiter par ici...

*même en tenant compte de nouvelles arrivées!*

## Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

## Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

53



## Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



16+

## Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



16+23+

## Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



16+23+51+

## Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

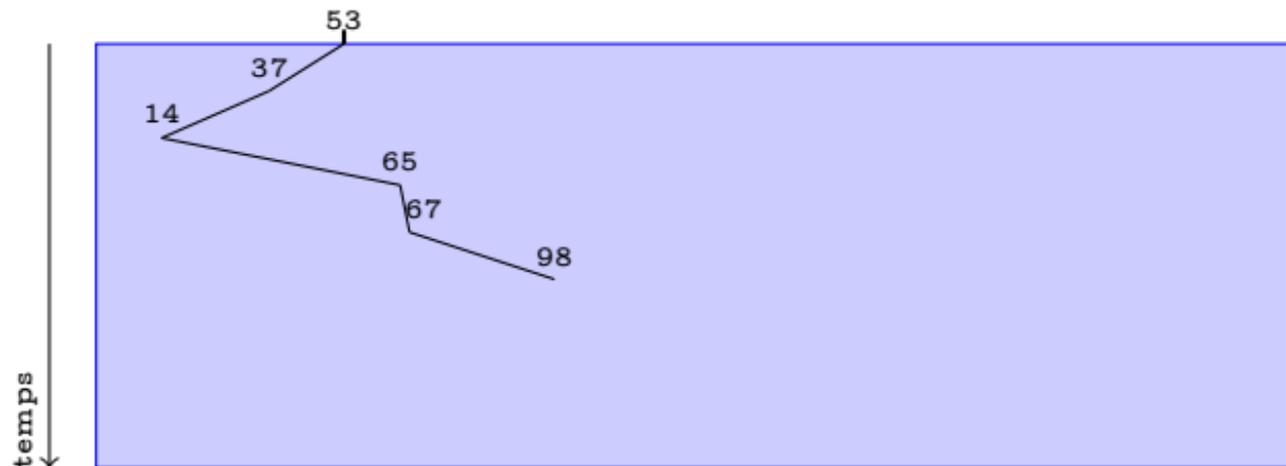


16+23+51+2+

## Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

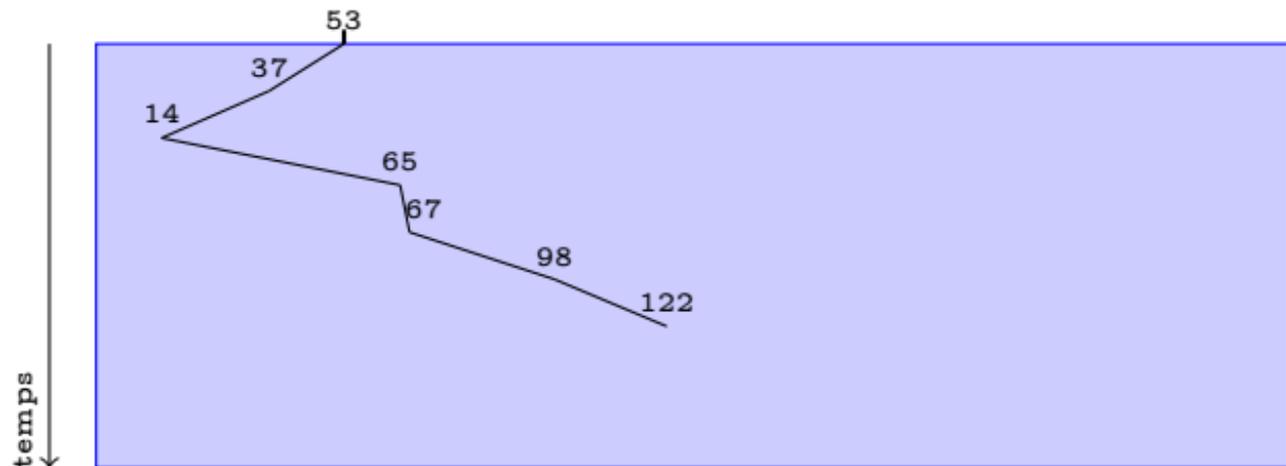


16+23+51+2+31+

## Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

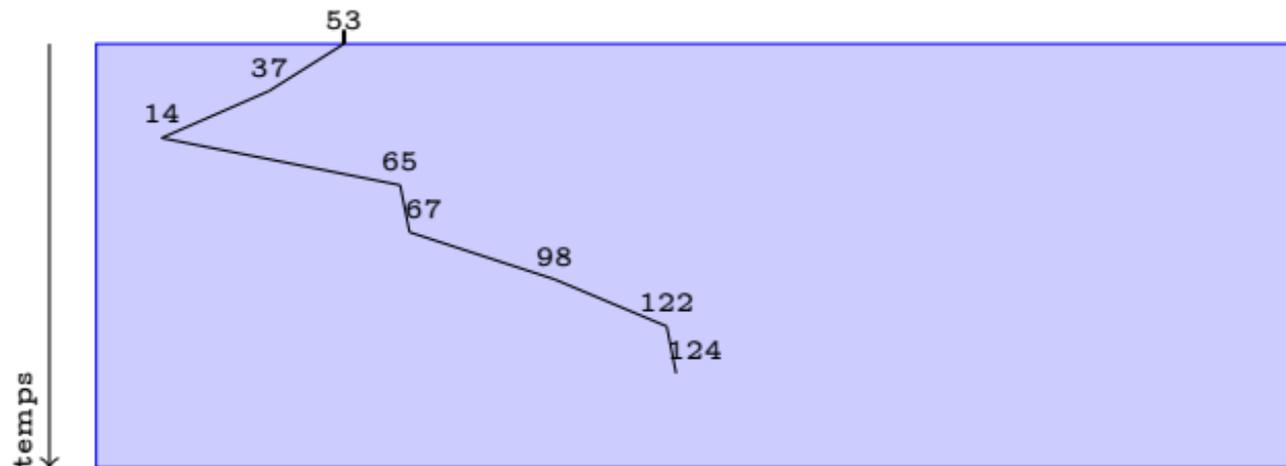


16+23+51+2+31+24+

## Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

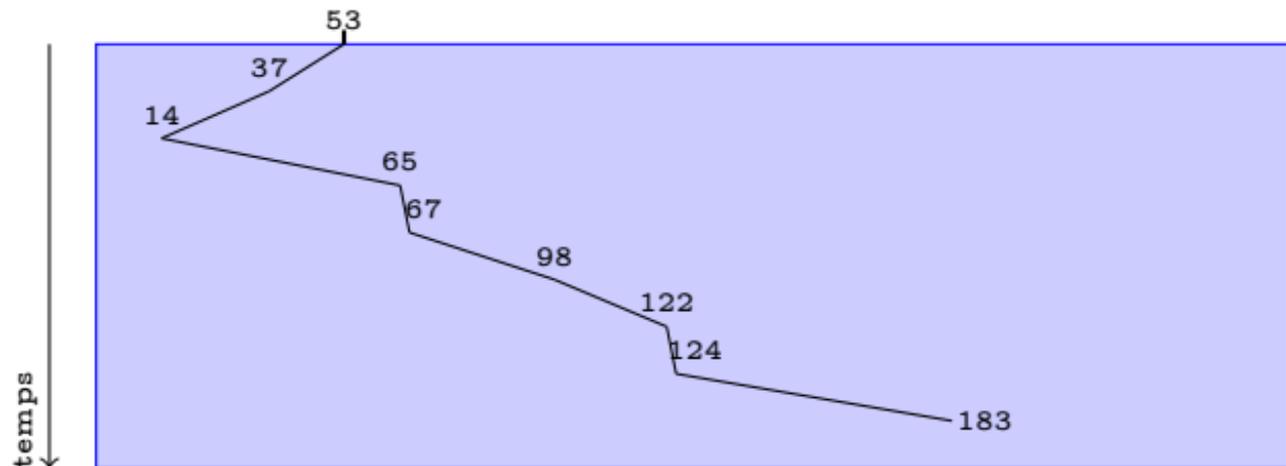


16+23+51+2+31+24+2+

## Principe

Repartir lorsqu'on a atteint le plus petit cylindre demandé

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



$$16+23+51+2+31+24+2+59 = (53-14) + (183-14) = 208$$

## Principe

Balayage circulaire : toujours dans le même sens

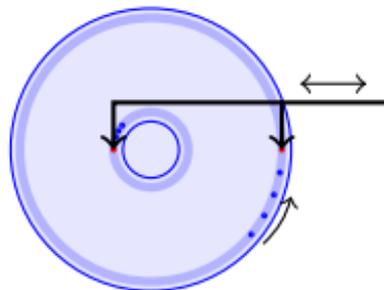
## Principe

Balayage circulaire : toujours dans le même sens

## Implémentation

Deux têtes de lecture espacées du rayon du disque

- Tête 1 lit cylindre 0 pendant que tête 2 lit le secteur n



## Principe

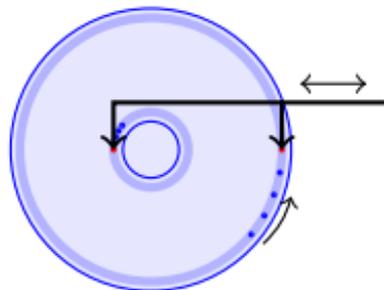
Balayage circulaire : toujours dans le même sens

## Implémentation

Deux têtes de lecture espacées du rayon du disque

- Tête 1 lit cylindre 0 pendant que tête 2 lit le secteur n
- Extérieur  $\rightarrow$  intérieur

Tête 1 balaye de 0 à n-1



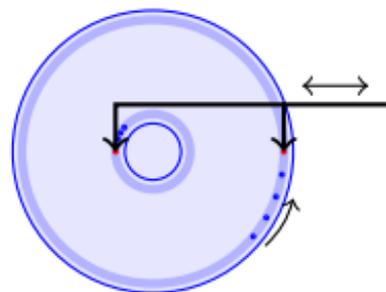
## Principe

Balayage circulaire : toujours dans le même sens

## Implémentation

Deux têtes de lecture espacées du rayon du disque

- Tête 1 lit cylindre 0 pendant que tête 2 lit le secteur n



- Extérieur  $\rightarrow$  intérieur

Tête 1 balaye de 0 à n-1

- Tête 1 = secteur n

$\rightarrow$  Tête 2 = secteur 0

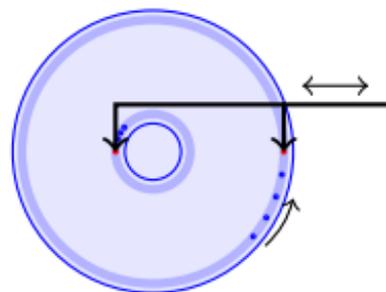
## Principe

Balayage circulaire : toujours dans le même sens

## Implémentation

Deux têtes de lecture espacées du rayon du disque

- Tête 1 lit cylindre 0 pendant que tête 2 lit le secteur n



- Extérieur  $\rightarrow$  intérieur

Tête 1 balaye de 0 à n-1

- Tête 1 = secteur n

$\rightarrow$  Tête 2 = secteur 0

- Intérieur  $\rightarrow$  extérieur

Tête 2 balaye de 0 à n-1

## Principe

Balayage circulaire (ici, descendant)

## Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

53

temps ↓



## Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



16+

## Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



16+23+

## Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



16+23+14+

## Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



16+23+14+

## Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

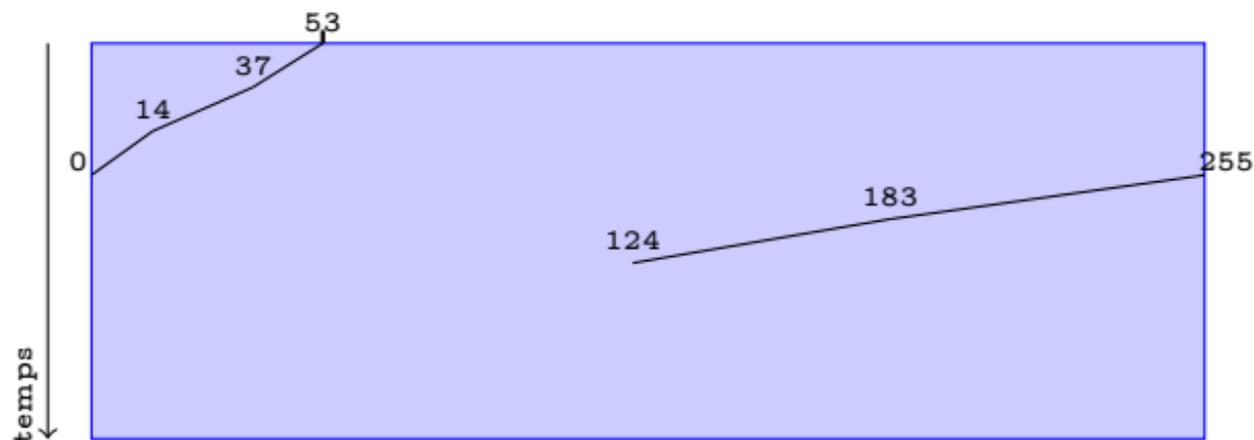


16+23+14+72+

## Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

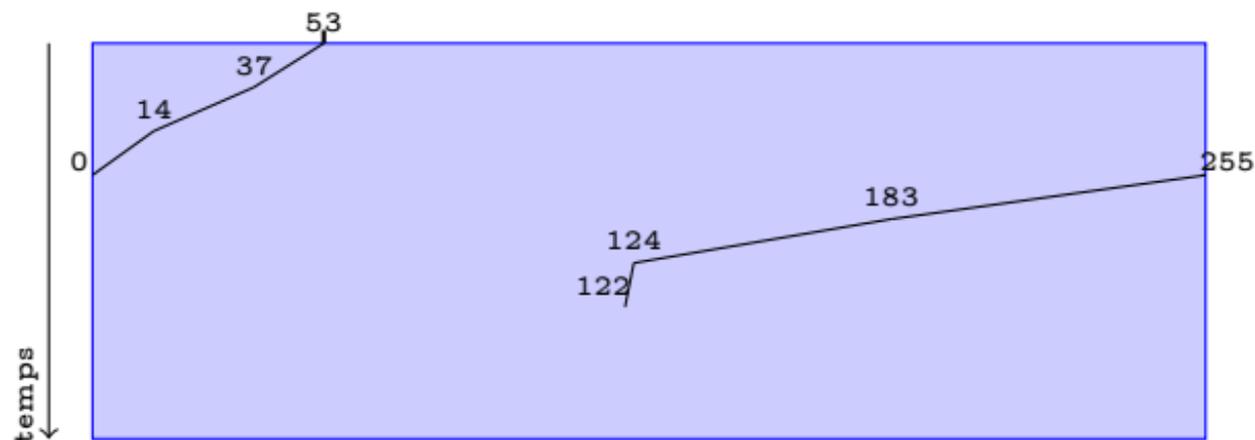


16+23+14+72+59+

## Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

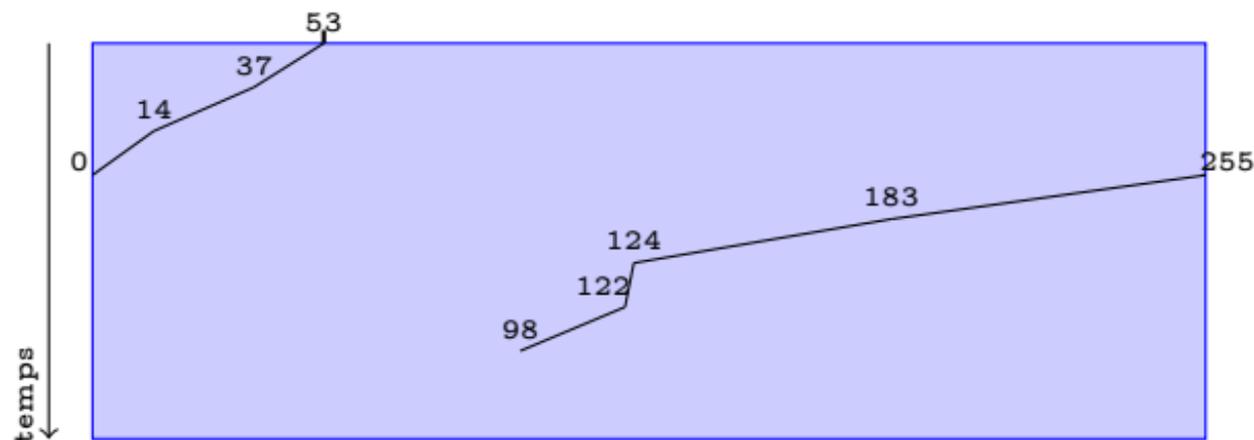


16+23+14+72+59+2+

## Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

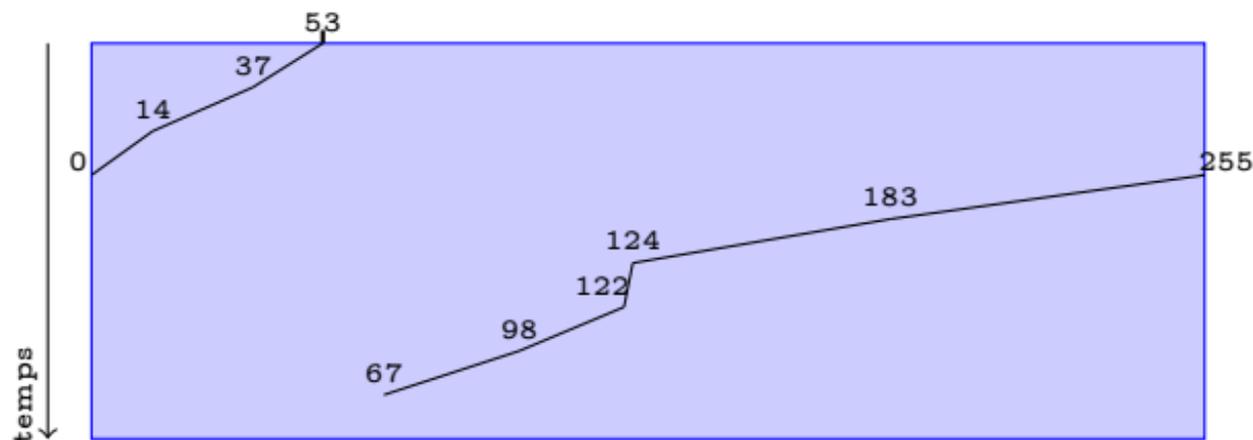


16+23+14+72+59+2+24+

## Principe

Balayage circulaire (ici, descendant)

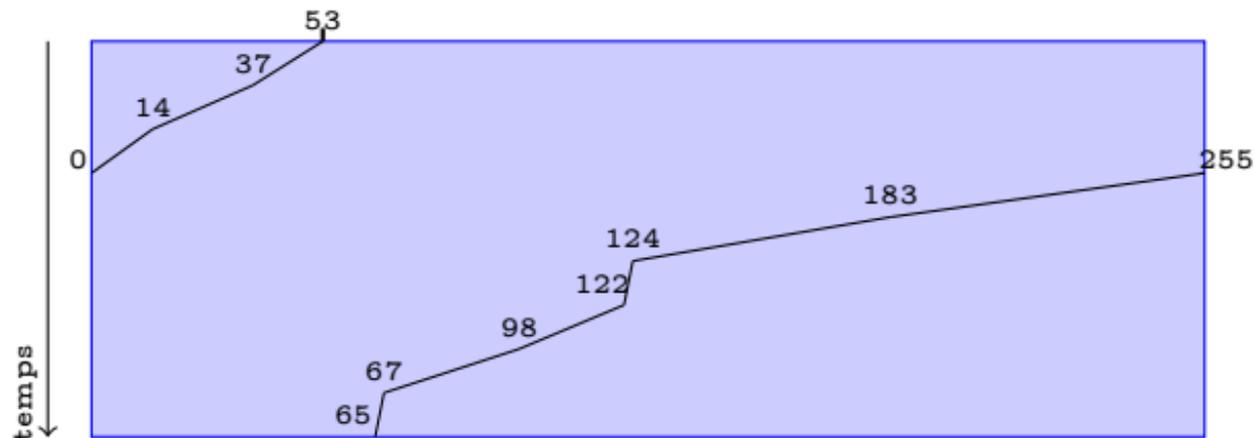
Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }


 $16+23+14+72+59+2+24+31+$

## Principe

Balayage circulaire (ici, descendant)

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



$$16+23+14+72+59+2+24+31+2 = 243$$

## Principe

Même principe sans aller jusqu'au bord

## Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

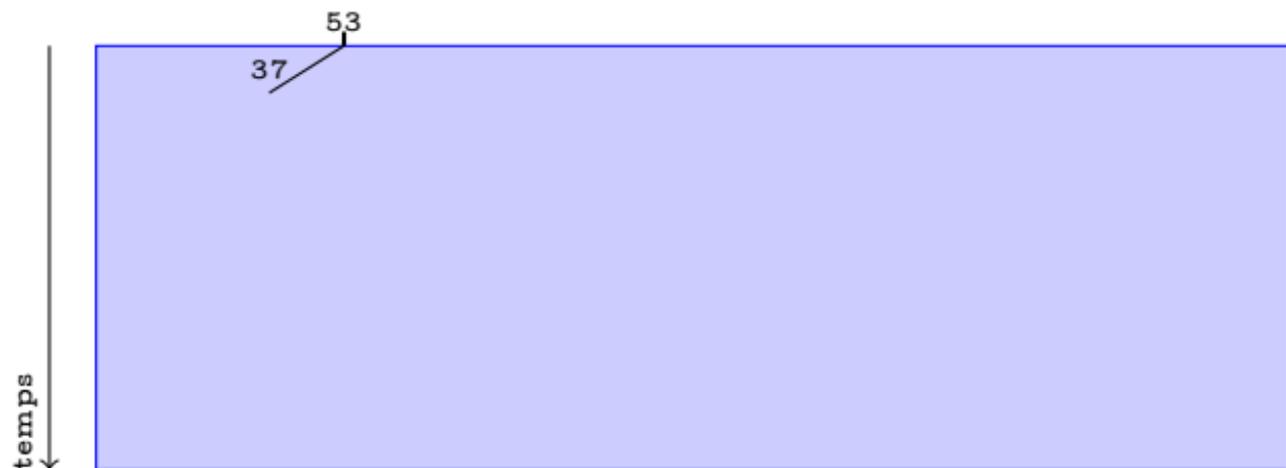
53



## Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

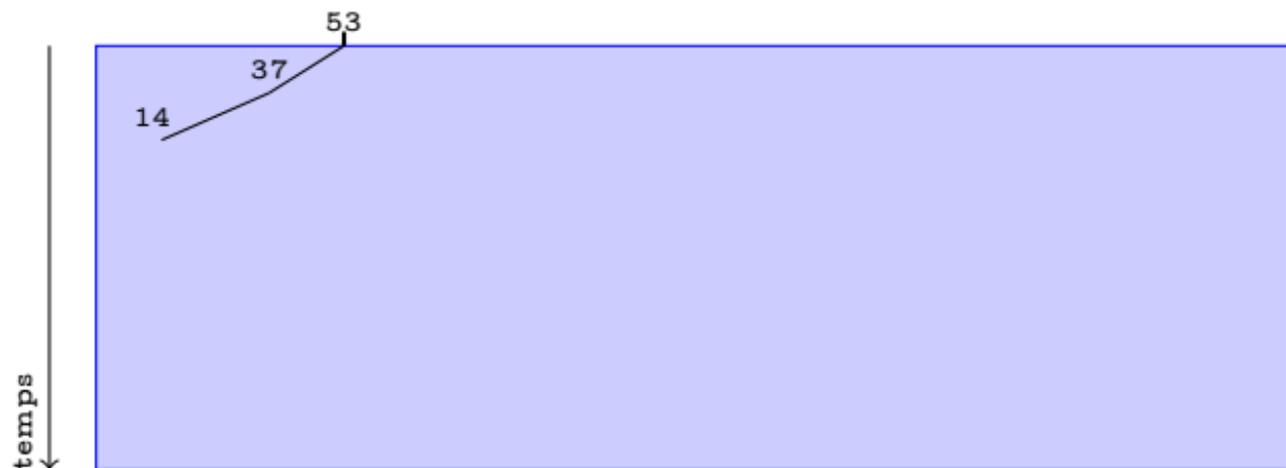


16+

## Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

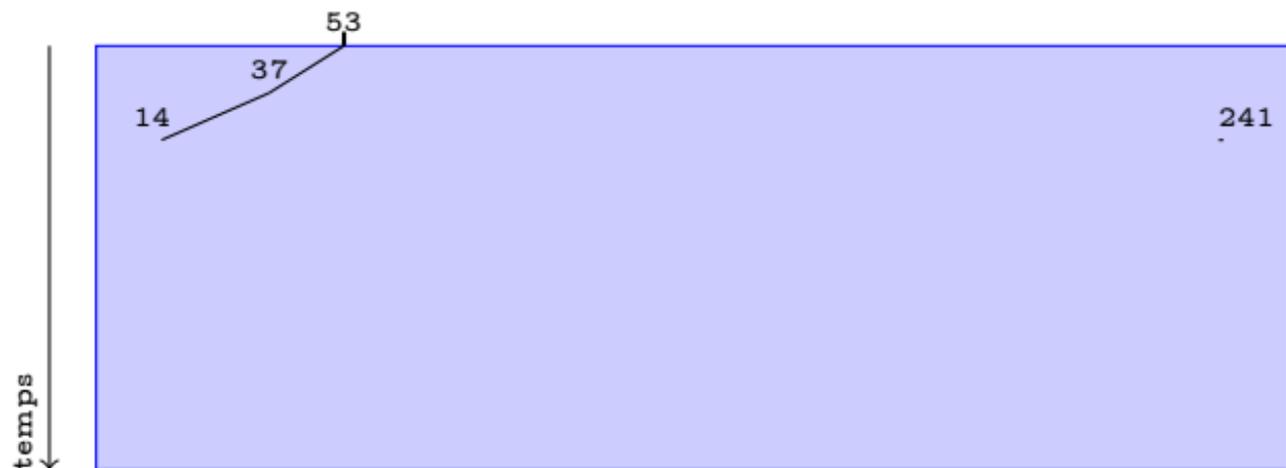


16+23+

## Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

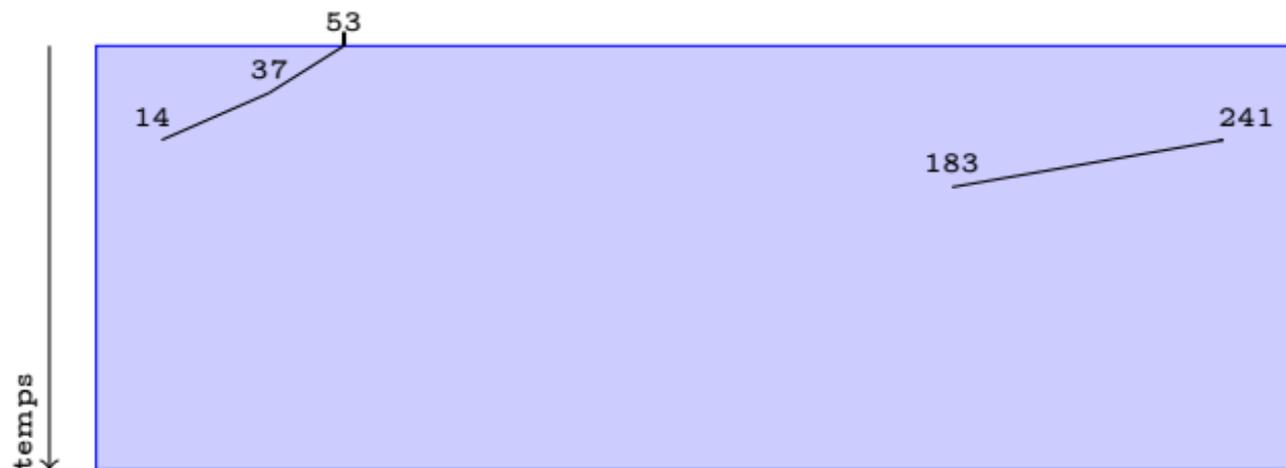


16+23+

## Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

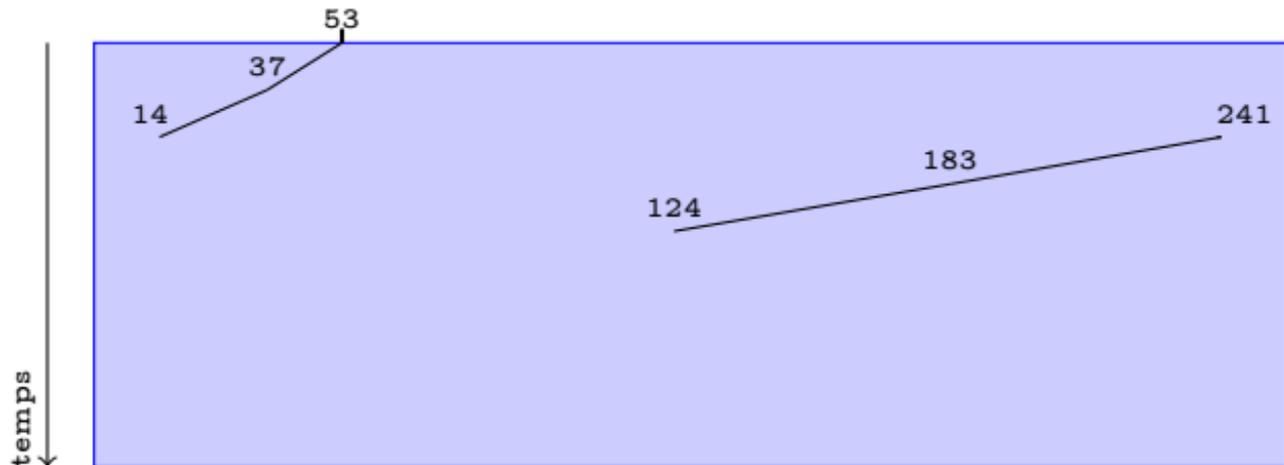


16+23+58+

## Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

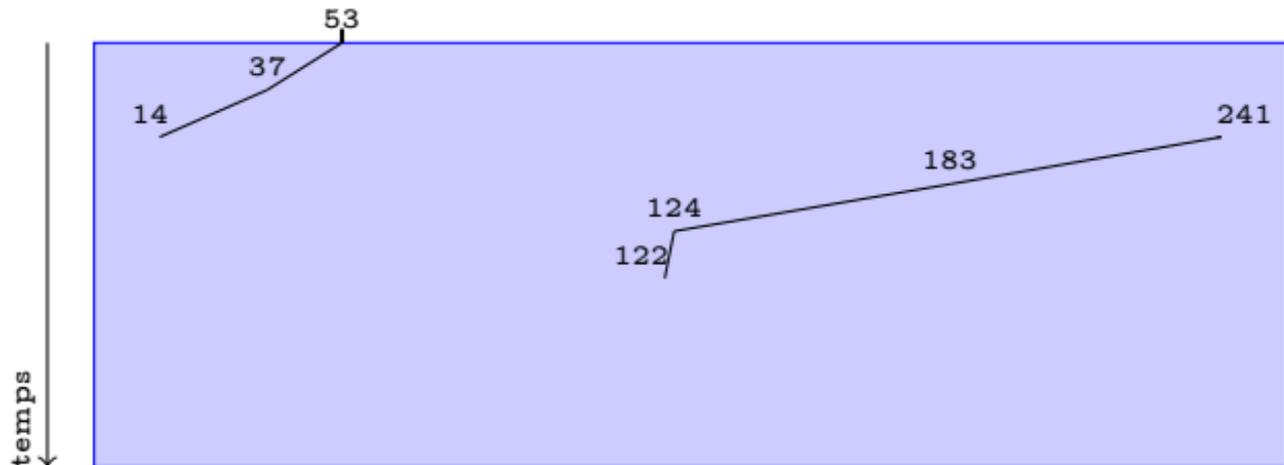


16+23+58+59+

## Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

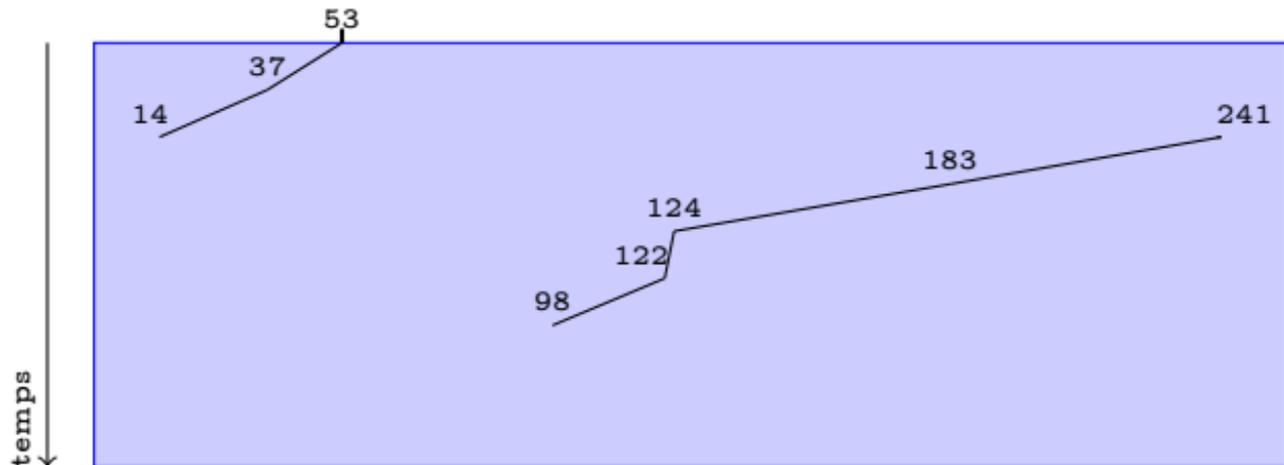


16+23+58+59+2+

## Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

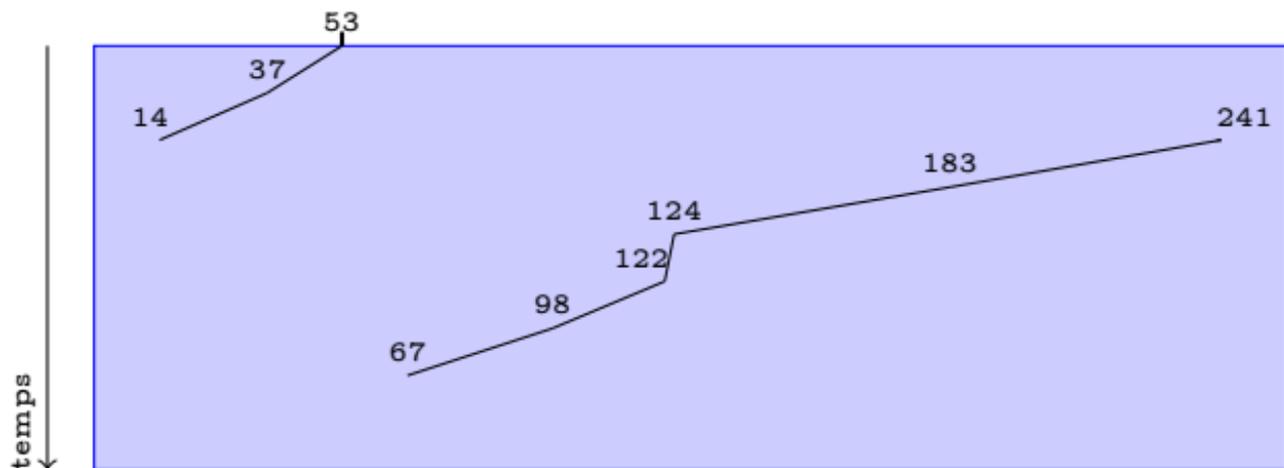


16+23+58+59+2+24+

## Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }

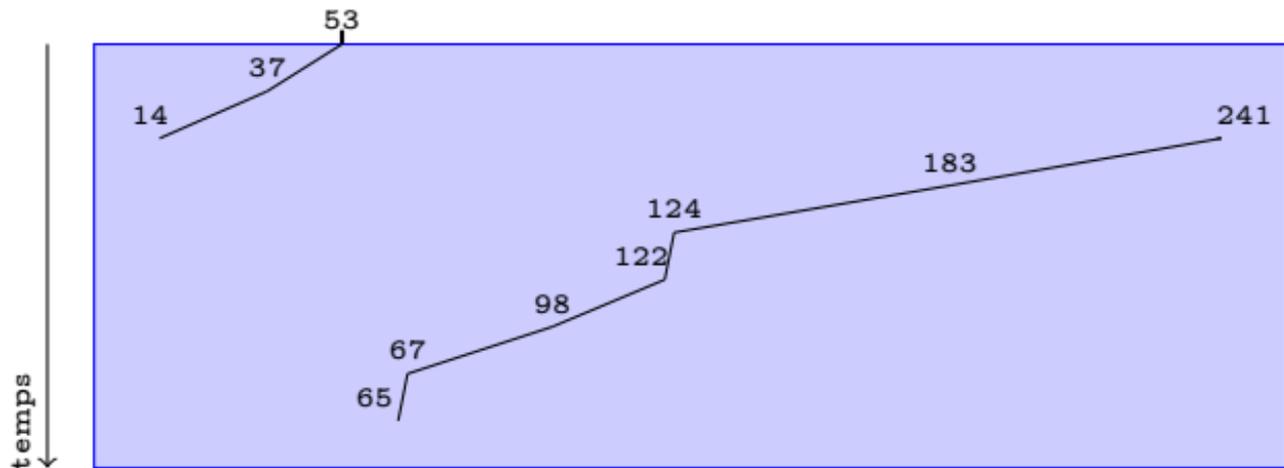


$$16+23+58+59+2+24+31+$$

## Principe

Même principe sans aller jusqu'au bord

Cylindre 53 + { 98, 183, 37, 122, 14, 124, 65, 67 }



$$16+23+58+59+2+24+31+2 = (53-14) + (241-65) = 215$$

### En pratique

- La plupart des OS utilisent SSTF
- Forte charge d'E/S : C-LOOK

### En pratique

- La plupart des OS utilisent **SSTF**
- Forte charge d'E/S : **C-LOOK**

### Ordonnancement optimal

Possible à calculer à chaque pas de temps mais très coûteux

## En pratique

- La plupart des OS utilisent **SSTF**
- Forte charge d'E/S : **C-LOOK**

## Ordonnancement optimal

Possible à calculer à chaque pas de temps mais très coûteux

## Rappel

Le temps de réponse dépend aussi de :

- La méthode d'**allocation de fichiers**
- La position des répertoires et des blocs d'index
- Le temps de rotation du disque
- La priorité au niveau OS (pagination vs E/S)

Bandes magnétiques

## Avantages

- ✓ Coût : **très peu cher** au Tio! (facteur 5 à 10)
  - Stockage préventif de données brutes
  - Ex : 15 To pour 40 EUR en 2020
- ✓ Coût de **maintenance** quasi nul
- ✓ **Durée de vie** (sans perte de données) plus élevée  
(CD = 5 ans, DD = 5 à 10 ans, Bande = 20 à 30 ans)
- ✓ Volume (ex : LTO-8 → 150 Go/cm<sup>3</sup>)

## Avantages

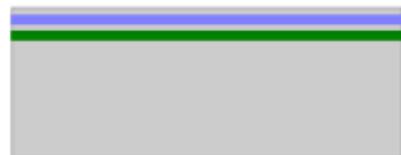
- ✓ Coût : **très peu cher** au Tio! (facteur 5 à 10)
  - Stockage préventif de données brutes
  - Ex : 15 To pour 40 EUR en 2020
- ✓ Coût de **maintenance** quasi nul
- ✓ **Durée de vie** (sans perte de données) plus élevée  
(CD = 5 ans, DD = 5 à 10 ans, Bande = 20 à 30 ans)
- ✓ Volume (ex : LTO-8 → 150 Go/cm<sup>3</sup>)

## Inconvénients

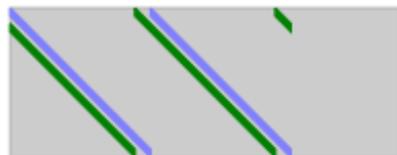
- ✗ Temps d'accès aléatoire et très élevé
- Réservé au stockage à froid ou application spécifiques

## Pistes

- 9 pistes : 8 données + 1 parité



*pistes linéaires*



*pistes hélicoïdales*

## Pistes

- 9 pistes : 8 données + 1 parité

## Blocs

- Secteurs taille fixe
- Intervalles inter-enregistrement (IRG)
  - entre les secteurs
  - On s'arrête uniquement sur les IRG
  - Interruption sur secteur
    - rembobiner à l'IRG précédent

## Pistes

- 9 pistes : 8 données + 1 parité

## Blocs

- Secteurs taille fixe
- Intervalles inter-enregistrement (IRG)
  - entre les secteurs
  - On s'arrête uniquement sur les IRG
  - Interruption sur secteur  
→ rembobiner à l'IRG précédent

## Avantages

- Lecture possible dans les 2 sens
- Stockage en baies