Skip lists - Listes à sauts

François Schwarzentruber

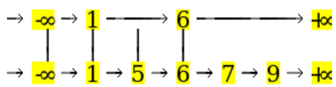
13 avril 2021

1 Définition

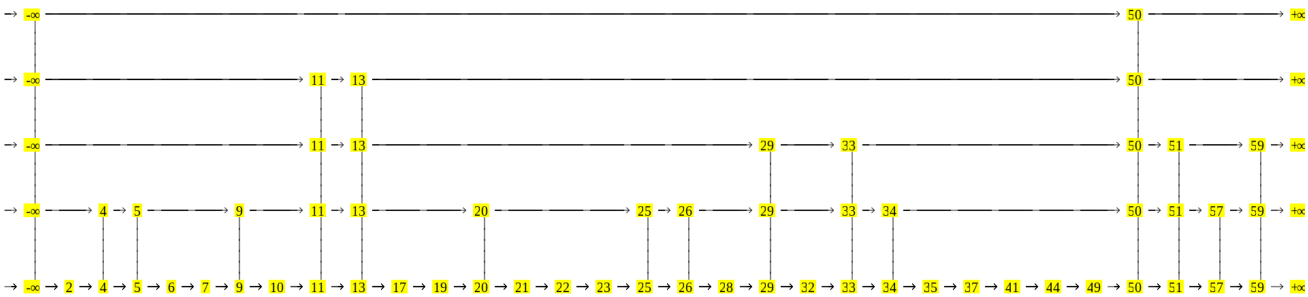
Vidéo 1 `https://www.youtube.com/watch?v=HIrh1NJmQxM`

Définition 2 (informelle) Une *skip list* est une liste simplement chaînée triée, avec des voies rapides (niveaux) en plus, pour prendre des raccourcis. Chaque niveau raffine le précédent. Le dernier niveau est la liste chaînée. Chaque nœud possède plusieurs pointeurs, un par voie. Le premier élément est une *sentinelle* et vaut $-\infty$.

Exemple 3 Voici une skip list avec 2 niveaux. Le niveau du bas est la liste chaînée habituelle, et le niveau du haut est la voie rapide.



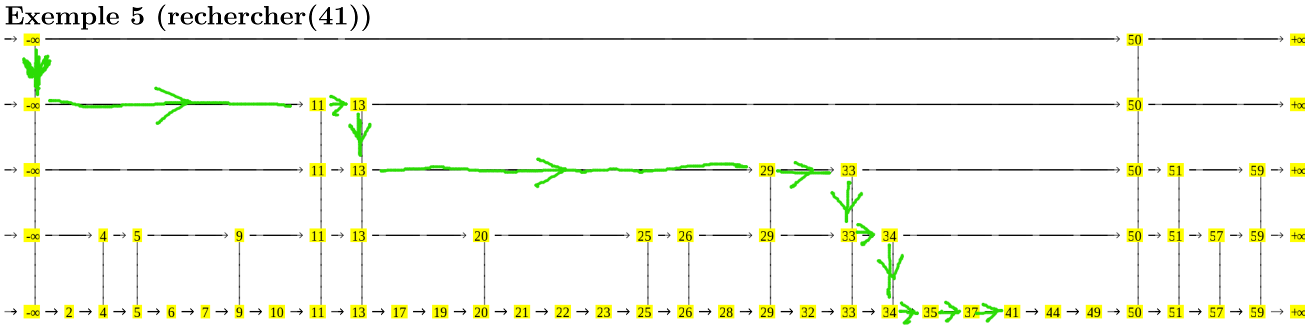
Exemple 4 Voici une skip list avec 5 niveaux.



2 Recherche

fonction rechercher(x)
- *skippy* := $-\infty$ sur le niveau le plus rapide (en haut à gauche)
- **tant que** *skippy* n'est pas sorti
  - avancer *skippy* vers la droite le plus possible sans dépasser v
  - **si** x trouvé **alors**
    - **renvoyer** oui
  - **sinon**
    - descendre *skippy* d'un niveau
- **renvoyer** non

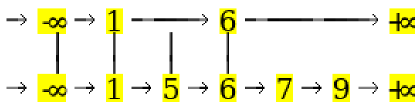
Exemple 5 (rechercher(41))



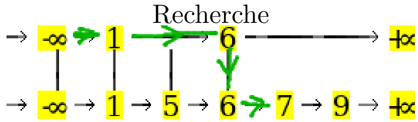
3 Ajout

```
fonction ajouter(x)
    aller à la position dans le niveau du bas où x doit être inséré (voir rechercher)
    y insérer x
    boucle
        lancer une pièce de monnaie
        si pile alors
            promouvoir x
        sinon
            s'arrêter
```

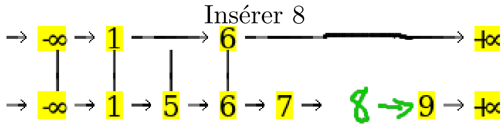
Exemple 6 (ajouter(8))



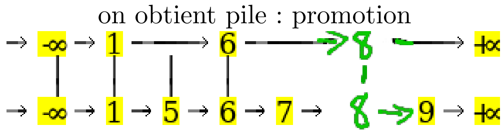
Recherche



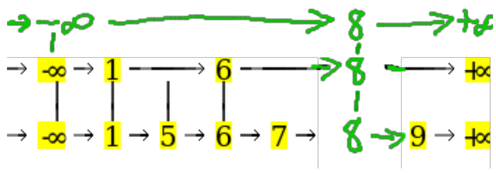
Insérer 8



on obtient pile : promotion



on obtient pile : promotion



on obtient face : on s'arrête

4 Meilleur équilibrage

Proposition 7 Pour le meilleur équilibrage d'une skiplist à k niveaux et n éléments, la complexité pire cas de la recherche est $O(kn^{1/k})$.

DÉMONSTRATION. Pour $k = 1$, le pire cas est linéaire.

Pour $k = 2$, le meilleur équilibrage revient à disposer les éléments pour découper le niveau du bas en part égal. En effet, si on note $|L1|$ la longueur de la liste du haut, et $|L0$ la liste du bas, il faut minimiser $|L1| + \frac{|L1|}{|L0|}$ où $|L1|$ est le temps passé dans la liste du haut et $\frac{|L1|}{|L0|}$ le temps passé dans la liste du bas. Le minimum est atteint quand les deux termes sont égaux. $|L1| = \frac{|L1|}{|L0|}$ donne $|L1| = \sqrt{n}$. D'où une complexité de $2\sqrt{n}$.

Pour trois niveaux, on obtient $3n^{1/3}$, etc. ■

Proposition 8 Pour le meilleur équilibrage d'une skiplist à $\log n$ niveaux et n éléments, la complexité pire cas de la recherche est $O(\log n)$.

DÉMONSTRATION. Pour $k = \log n$, $O(kn^{1/k})$ vaut $O(\log n)$! En effet, $\log n n^{1/\log n} = \log n 2^{\frac{\log n}{\log n}} = 2 \log n$. ■

5 Rappels de probabilité

Proposition 9 Soit X une variable aléatoire à valeurs dans $\mathbb{N}$. On a : $\mathbb{E}(X) = \sum_{k=1}^{\infty} \mathbb{P}(X \geq k)$.

DÉMONSTRATION. On réécrit la définition de l'espérance :

$$\begin{aligned}
\mathbb{E}(X) &= \sum_{k=1}^{\infty} k\mathbb{P}(X{=}k) \\
&= \begin{array}{rrr}
\mathbb{P}(X{=}1){+}\mathbb{P}(X{=}2){+} & \mathbb{P}(X{=}3){+}\mathbb{P}(X{=}4){+} & \mathbb{P}(X{=}5)+\dots \\
\mathbb{P}(X{=}2){+} & \mathbb{P}(X{=}3){+}\mathbb{P}(X{=}4){+} & \mathbb{P}(X{=}5)+\dots \\
 & \mathbb{P}(X{=}3){+}\mathbb{P}(X{=}4){+} & \mathbb{P}(X{=}5)+\dots \\
 & \mathbb{P}(X{=}4){+} & \mathbb{P}(X{=}5)+\dots \\
 & & \mathbb{P}(X{=}5)+\dots
\end{array} \\
&= \sum_{k=1}^{\infty}\sum_{j=k}^{\infty} \mathbb{P}(X{=}j) \\
&= \begin{array}{l}
\mathbb{P}(X{\geq}1){+} \\
\mathbb{P}(X{\geq}2){+} \\
\mathbb{P}(X{\geq}3){+} \\
\mathbb{P}(X{\geq}4){+} \\
\mathbb{P}(X{\geq}5){+}\dots
\end{array} \\
&= \sum_{k=1}^{\infty} \mathbb{P}(X \geq k)
\end{aligned}$$

■

Proposition 10 (loi géométrique de paramètre $\frac{1}{2}$) Soit X le nombre de piles consécutifs en lançant une pièce de monnaie équilibrée. On a : $\mathbb{E}(X) = 1$.

DÉMONSTRATION. $\mathbb{E}(X) = \sum_{k=1}^{\infty} \mathbb{P}(X \geq k) = \sum_{k=1}^{\infty} \frac{1}{2^k} = 1$. ■

Définition 11 (espérance conditionnelle) Soit X une variable aléatoire discrète et E un événement de probabilité non nulle.

$$\mathbb{E}(X \mid E) = \frac{1}{\mathbb{P}(E)} \sum_{k=1}^{\infty} k\mathbb{P}(X{=}k \ \cap \ E).$$

Proposition 12 Si $A \sqcup B = \Omega$ (partition) alors $\mathbb{E}(X) = \mathbb{E}(X \mid A)\mathbb{P}(A) + \mathbb{E}(X \mid B)\mathbb{P}(B)$.

6 Complexité

Proposition 13 Le nombre de niveau d'une skip list avec n éléments est $O(\log n)$ avec une grande probabilité. Plus précisément, pour tout $c \geq 1$, $\mathbb{P}(\text{nb de niveaux} > c\log n) \leq \frac{1}{n^{c-1}}$.

DÉMONSTRATION.

$$\begin{aligned}
\mathbb{P}(\text{nb de niveaux} > c\log n) &= \mathbb{P}(\text{au moins un des éléments est promu} \geq c\log n \text{ fois}) \\
&\leq n \times \mathbb{P}(\text{un élément particulier est promu} \geq c\log n \text{ fois}) && \text{proba d'une union} \\
&= n\left(\frac{1}{2}\right)^{\lfloor c\log n \rfloor} && \lfloor c\log n \rfloor \text{ faces} \\
&\leq n\left(\frac{1}{2}\right)^{c\log n} \\
&\leq n\left(\frac{1}{(2^{\log n})^c}\right) && \text{calcul} \\
&= \frac{n}{n^c} = \frac{1}{n^{c-1}} && \text{calcul}
\end{aligned}$$

■

Corollaire 14 $\mathbb{E}(\text{nb de niveaux}) = O(\log n)$.

Démonstration. On utilise le fait que $\mathbb{E}(X) = \sum_k \mathbb{P}(X \geq k)$, puis on découpe la somme par paquet de taille $\log n$.

$$\begin{aligned}
\mathbb{E}(X) &= \sum_{k=1}^{\infty} \mathbb{P}(X \geq k) \\
&= \sum_{k=1}^{\lfloor \log n \rfloor} \mathbb{P}(X \geq k) + \sum_{k=1+\lfloor \log n \rfloor}^{\lfloor 2\log n \rfloor} \mathbb{P}(X \geq k) + \sum_{k=1+\lfloor 2\log n \rfloor}^{\lfloor 3\log n \rfloor} \mathbb{P}(X \geq k) + \dots \\
&\leq \log n \times 1 + \log n \times \mathbb{P}(X > \log n) + \log n \times \mathbb{P}(X > 2\log n) + \dots \\
&\leq \log n \times (1 + 1 + \frac{1}{n} + \frac{1}{n^2} + \frac{1}{n^3} + \dots) \text{ en utilisant la proposition 13} \\
&\leq \log n \times \left(1 + \frac{1}{1 - \frac{1}{n}}\right) = O(\log n)
\end{aligned}$$

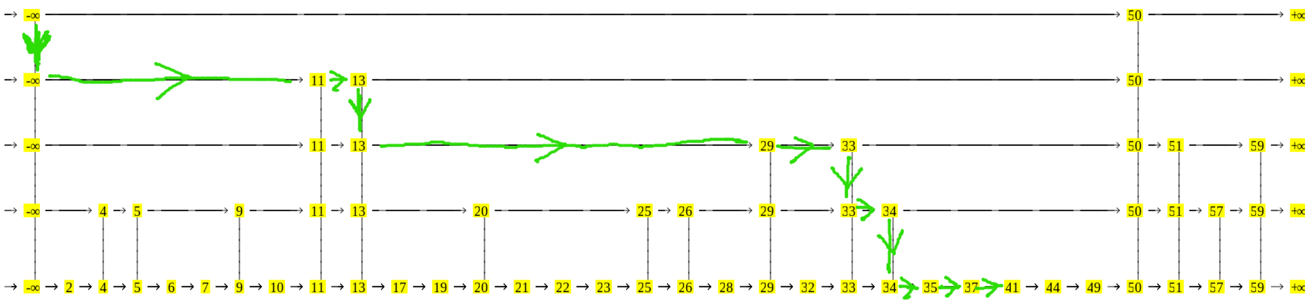
■

Proposition 15 L'espérance de la mémoire utilisée est $O(n)$.

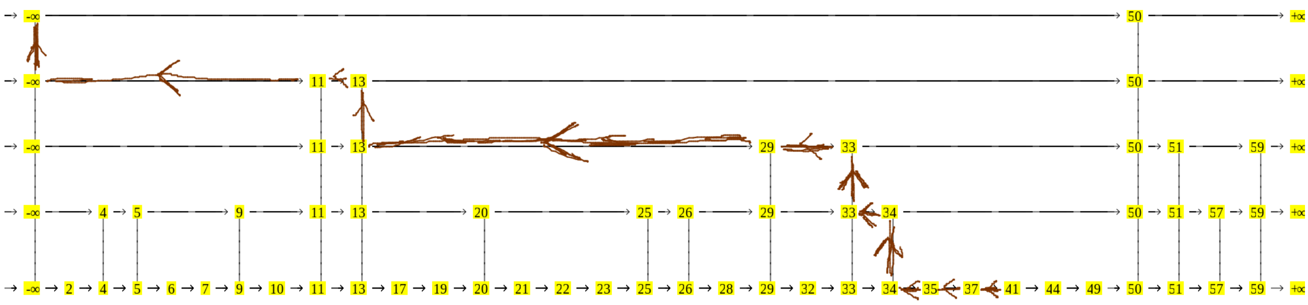
Démonstration. La mémoire utilisée par un élément donné est sa hauteur H, i.e. $1 +$ nb de fois qu'il a été promu. La loi de l'espace mémoire utilisée par un élément est le même. Donc l'espérance de la mémoire totale utilisée est $n\mathbb{E}(H)$. La loi de H est géométrique, mais commence à 1. L'espérance de H est donc de $1 + \frac{1}{2}$, d'où le résultat. ■

Proposition 16 Une recherche dans une skip list coûte $O(\log n)$ en temps espéré.

Démonstration. Posons C le nombre de nœuds visités durant la recherche. A constante près, C est la complexité de la recherche. Soit C_i le nombre de nœuds visités au niveau numéro i. En effet, considérons une exécution d'une recherche. Elle se lit comme un chemin dans la structure composé de $\rightarrow$ et de $\downarrow$.



Lisons ce chemin à l'envers à partir de l'élément recherché jusqu'au $-\infty$ en haut à gauche.



Sur ce chemin inverse, on prend un $\uparrow$ à rebours parce que l'élément courant a été promu, i.e. vaut $\frac{1}{2}$. Ainsi, l'espérance du nombre de nœuds visités à un certain niveau est inférieure à 2. Autrement dit $\mathbb{E}(C_i) \leq 2$.

On a :

$$\begin{aligned}
\mathbb{E}(C_i) &= \mathbb{E}(C_i \mid C_i = 0)\mathbb{P}(C_i = 0) + \mathbb{E}(C_i \mid C_i \geq 1)\mathbb{P}(C_i \geq 1) \\
&= 0 \times \mathbb{P}(C_i = 0) + \mathbb{E}(C_i \mid C_i \geq 1)\mathbb{P}(C_i \geq 1) \\
&= 0.
\end{aligned}$$

Comme il y a des nœuds traités au niveau i ssi il y a plus de i niveaux, on a :

$$\mathbb{P}(\text{nb de niveaux} \geq i) = \mathbb{P}(C_i \geq 1).$$

On a :

$$\begin{aligned}
\mathbb{E}(C) &= \sum_{i=1}^{\infty} \mathbb{E}(C_i) \\
&= \sum_{i=1}^{\infty} \mathbb{E}(C_i \mid C_i \geq 1)\mathbb{P}(C_i \geq 1) \\
&= \sum_{i=1}^{\infty} \mathbb{E}(C_i \mid C_i \geq 1)\mathbb{P}(\text{nb de niveaux} \geq i) \\
&\leq 2\sum_{i=1}^{\infty} \mathbb{P}(\text{nb de niveaux} \geq i) \\
&= \sum_{k=1}^{\lfloor \log n \rfloor} \mathbb{P}(\text{nb de niveaux} \geq k) + \sum_{k=1+\lfloor \log n \rfloor}^{\lfloor 2\log n \rfloor} \mathbb{P}(\text{nb de niveaux} \geq k) + \sum_{k=1+\lfloor 2\log n \rfloor}^{\lfloor 3\log n \rfloor} \mathbb{P}(\text{nb de niveaux} \geq k) + \dots \\
&= O(\log n)(1 + 1 + \frac{1}{n} + \frac{1}{n^2} + \dots) \\
&= O(\log n).
\end{aligned}$$

.

■

7 Encore des rappels de probabilité

Théorème 17 (inégalité de Chernoff) Soit Y le nombre total de faces dans une suite de m lancées de pièces. Pour tout $r > 0$,

$$\mathbb{P}(Y \geq \frac{m}{2} + r) \leq e^{-2r^2/m}.$$

Démonstration. Voir un cours de probabilité. ■

8 Complexité avec grande probabilité

Lemme 18 Pour tout constante c, il existe d tel que, avec grande probabilité, le nombre de piles en lançant $d \log n$ pièces est au moins $c \log n$. Plus formellement :

$$\mathbb{P}(\text{nb de piles obtenus durant au moins } d\log n \text{ lancés} \geq c \log n) \quad \geq \quad 1 - \frac{1}{n^c}.$$

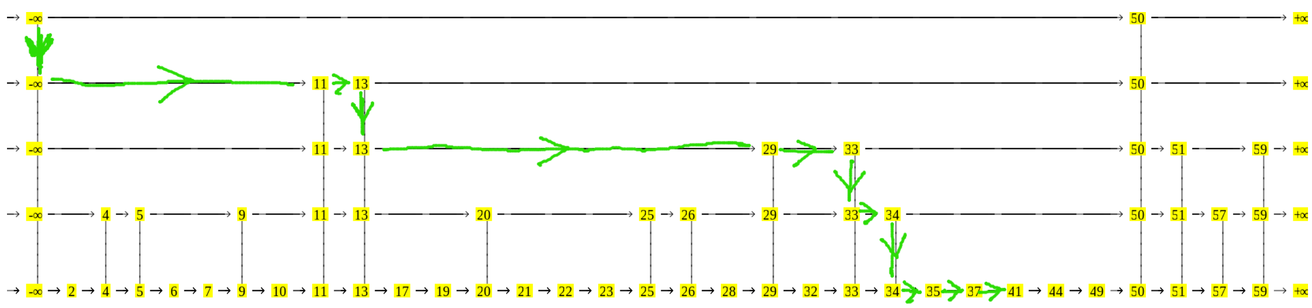
Démonstration.
Soit Y le nb de faces avec $d \log n$ lancés.

$$\mathbb{P}(\leq c\log n \text{ piles}) = \mathbb{P}(Y \geq d\log n - c\log n) = \mathbb{P}(Y \geq m/2 + r).$$

On a $m = d\log n$ et on pose $r = (d/2 - c)\log n$. On reconnait donc : $\mathbb{P}(Y \geq m/2 + r)$. Cette probabilité est donc $\leq e^{-2r^2/m}$. par l'inégalité de Chernoff. Par le calcul, en posant $d = 3c$, on arrive à montrer qu'elle est plus petite que $\frac{1}{n^c}$. ■

Proposition 19 Une recherche dans une skip list coûte $O(\log n)$ avec une grande probabilité.

Démonstration. Considérons un certain c, et la constante d du lemme 18. Considérons une recherche et soit L = la longueur du chemin de Skippy :



Événement A : nb de $\downarrow \leq c \log n$.
Événement B : $L \leq d \log n$

Le but est de montrer que $\mathbb{P}(B)$ grande, i.e. que $\mathbb{P}(\text{non } B)$ petite. On a :

$$\begin{aligned} \mathbb{P}(\text{not } (B)) &\leq \mathbb{P}((\text{non } A) \text{ ou } (\text{non } B \text{ et } A)) \\ &\leq \mathbb{P}(\text{non } A) + \mathbb{P}(A \text{ et non } B) \end{aligned}$$

nombre de $\downarrow$ = nombre de piles le long du chemin
Quand A se produit, L = nombre de lancées de pièces pour avoir strictement plus de $c \log n$ pile.

$$\frac{\mathbb{P}(\text{non } A) \leq \mathbb{P}(\text{nb de niveaux} > c\log n) \qquad \dfrac{\text{Proposition 13}}{\mathbb{P}(\text{non } A) \geq \mathbb{P}(\text{nb de niveaux} > c\log n) \leq \frac{1}{n^{c-1}}}}{\mathbb{P}(\text{non } A) \leq \frac{1}{n^{c-1}}}$$

$$\frac{\text{nombre de } \downarrow = \text{nombre de piles le long du chemin} \qquad \text{Lemme 18}}{\mathbb{P}((\text{non } B \text{ et } A)) \leq \frac{1}{n^c}}$$

Ainsi

$$\mathbb{P}(\text{not } (B)) \leq \frac{1}{n^{c-1}} + \frac{1}{n^c} = O(\frac{1}{n^{c-1}}).$$

■

9 Implémentation en Python

https://github.com/francoisschwarzentruber/skiplist/blob/main/skiplist.py

10 Notes bibliographiques

Les skip lists ont été inventées par William Pugh [Pug90]. Elles ont inspiré les *skip graphs* utilisés en réseau pair-à-pair [AS07]. Vous trouvez des informations pour améliorer l'implémentation ici : [BFJ+17].

Références

[AS07] James Aspnes and Gauri Shah. Skip graphs. *ACM Trans. Algorithms*, 3(4) :37, 2007.

[BFJ+17] Michael A. Bender, Martin Farach-Colton, Rob Johnson, Simon Mauras, Tyler Mayer, Cynthia A. Phillips, and Helen Xu. Write-optimized skip lists. In Emanuel Sallinger, Jan Van den Bussche, and Floris Geerts, editors, *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, pages 69–78. ACM, 2017.

[Pug90] William Pugh. Skip lists : A probabilistic alternative to balanced trees. *Commun. ACM*, 33(6) :668–676, 1990.