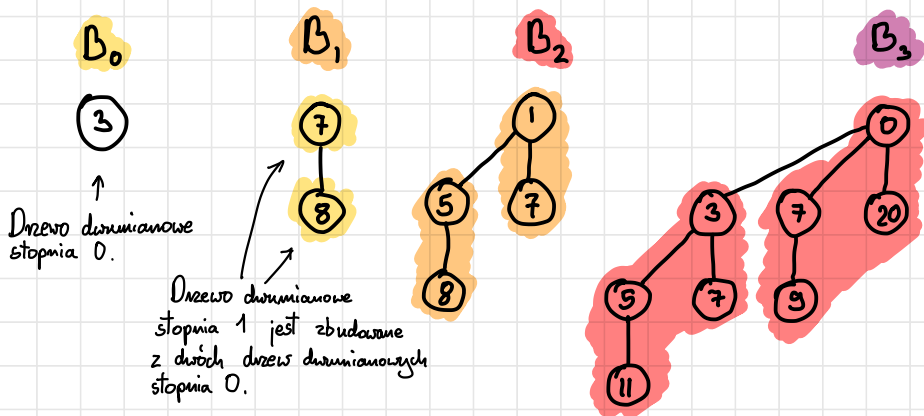


Kopce Dwumianowe,

Kopce Drzewianowe (kontinue)

Są to kopce, które mają strukturę drzewiastą, ale zasady kopcowe.

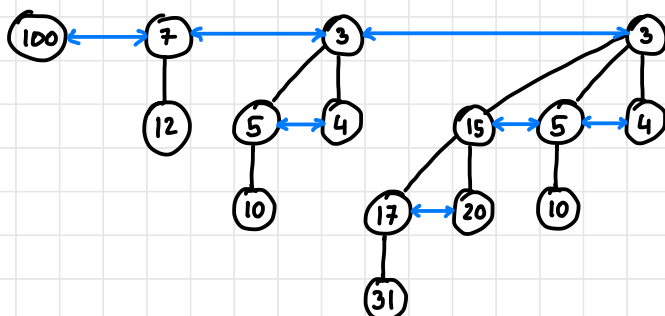
Kopiec drzewianowy jest to double linked list z drzewami drzewianowymi, czyli takimi, które są zbudowane z dwóch drzew drzewianowych stopnia 1 mniej. Jedno jest umieszczone w korzeniu, a drugie jest dołączone jako lewe dziecko pierwszego.



Kopiec drzewianowy to taki kopiec na którym każde dziecko tego samego wierzchołka jak i wszystkie korzenie są połączone razem za pomocą double linked list.

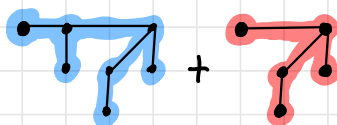
- Każde drzewo drzewianowe w kopcu ma unikalny stopień k .
- Stopień korzenia wynosi k (ma k dzieci).
- Wysokość drzewa B_k wynosi k .
- Jest 2^k wierzchołków w każdym drzewie.

Budowa kopca dwumianowego

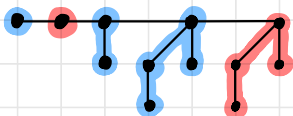


Scalanie

Scalmy

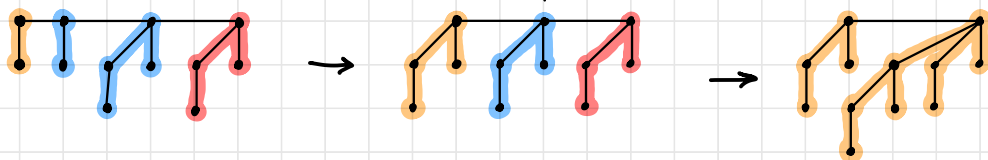


1. Wtłuczamy te dwa kopce ze sobą



2. Iterujemy się od lewej strony i scalamy drzewa gdy napotkamy duplikaty (jeśli pojawia się 3 drzewa jako skutek uboczny z poprzedniej iteracji, to scala dwa ostatnie).

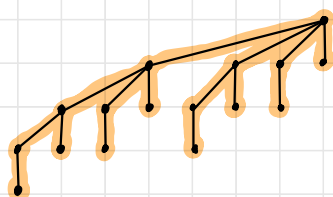
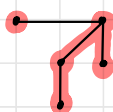
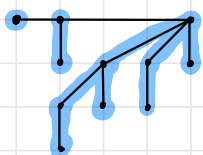
Przypadek z nawiasu



Liczeby binarne

Każdy kopiec możemy opisać liczbami binarnymi:

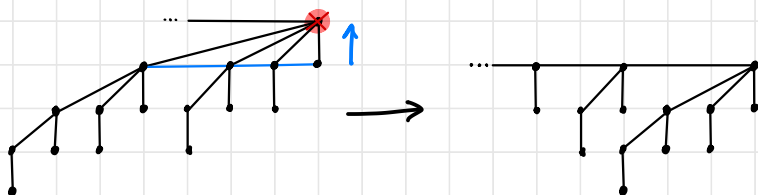
$$\begin{array}{cccc} 1 & 1 & 0 & 1 \\ b_0 & b_1 & b_2 & b_3 \end{array} (11) + \begin{array}{ccc} 1 & 0 & 1 \\ b_0 & b_1 & b_2 \end{array} (5) = \begin{array}{ccccc} 0 & 0 & 0 & 0 & 1 \\ b_0 & b_1 & b_2 & b_3 & b_4 \end{array} (16)$$



Każda propagacja bitu, to jedno scalecie pojedynczego drzewa.

Delete min

Wykorzystujemy fakt, że dzieci wierzchołka są w liście wiazanej.



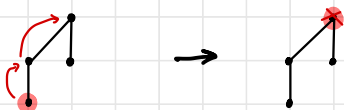
Usuwamy wierzchołek i przenosimy dzieci wyżej w odwrotnej kolejności.

Operacje

min	$O(\log n)$
insert	$O(1)^*$
merge	$O(\log n)$
delete min	$O(\log n)$

Delete

Jeśli usuwamy element z drzewa, to przepychamy go na wierzch, a następnie wykonujemy delete min.



Złożoność czasowa zaimplementowana dla insert *

Dodajemy n elementów do pustego kopca.
Policzmy ile scalań s musimy za każdym razem wykonać

i	1	2	3	4	5	6	7	8	9	10	...
s	0	1	0	2	0	1	0	3	0	1	...

Jak możemy zauważyć co drugi element jest chociaż raz skalany, co czwarty kolejny raz itd.

Ma to związek z reprezentacją binarną kopca.

Zatem:

$$\frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \dots + \frac{n}{n} \approx 2n \text{ dla } n \text{ wyrotacji}$$

Zatem średnio dla 1 wyrotania ok. 2 operacje

$O(1)$ amortyzowanej złożoności czasowej.

Kopiec dwumianowy (leniwy)

- Prokrastynuje i zostawia robotę na potem.
- Porządkuje drzewo dopiero przy delete min
- To oznacza, że może mieć więcej niż jedno B_k .

Operacje amortyzowane

min	$O(1)$	- zapamiętuje minimum
insert	$O(1)$	- po prostu dodaje element do listy związanej
merge	$O(1)$	- po prostu łączy drzewa (eventualnie aktualizuje min)
delete min	$O(\log n)$	- usuwa min i sprząta kopiec (porządkuje zapamiętane min w drzewach)