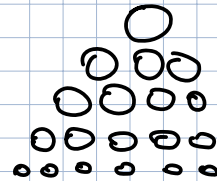


Пирамида

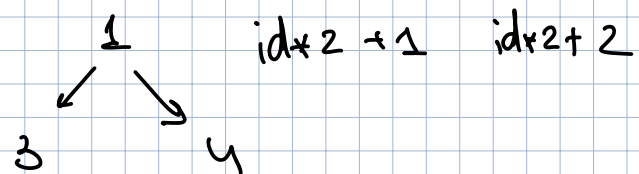
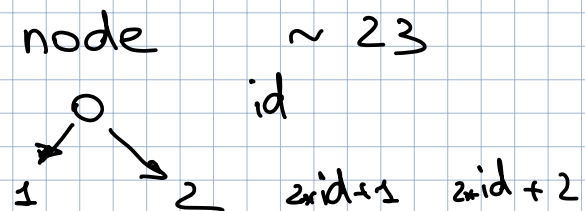
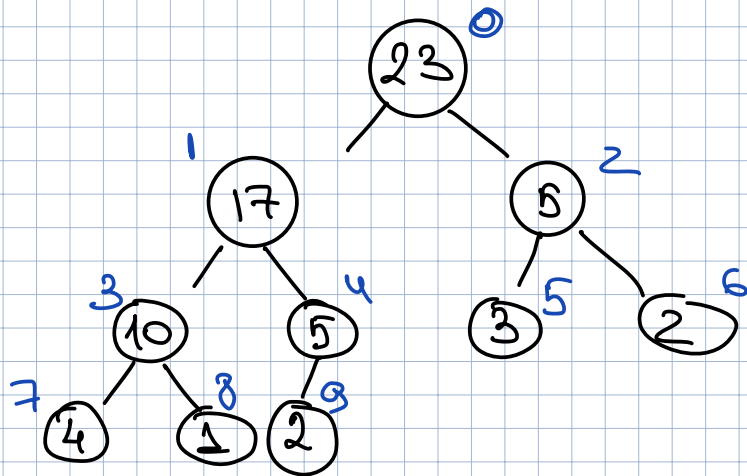
АТД кута (очередь с приоритетом)
Heap

Запоминка :



Главное правило:
чем выше, тем больше.

Пример:



„Левый” ребенок: $id * 2 + 1$

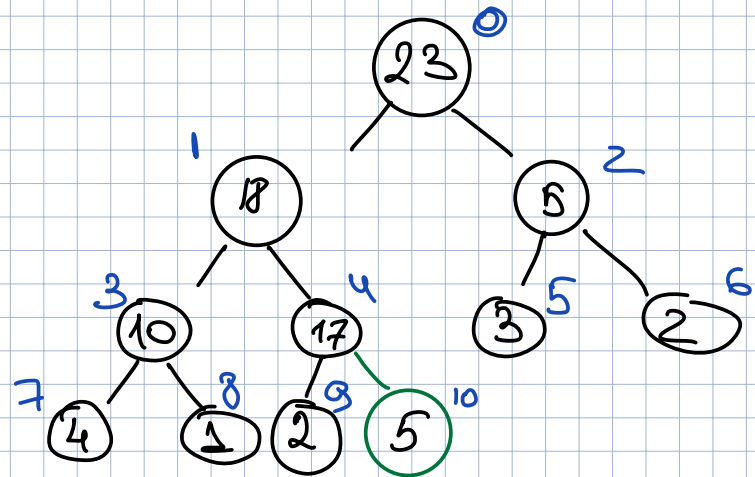
„Правый” ребенок: $id * 2 + 2$

$$\text{id-child} \rightarrow \text{id-parent} = \left\lfloor (\text{id-child} - 1) / 2 \right\rfloor$$

Проход с нуля 1

Добавил один
"неподходящий" элемент.

бросил



Sift Up (arr, id) {

while (id > 0)

{

id-parent = (id-1) / 2

если

arr[id-parent] < arr[id]

↳ swap(arr[id-parent], arr[id])

id = id-parent

return

} } → break

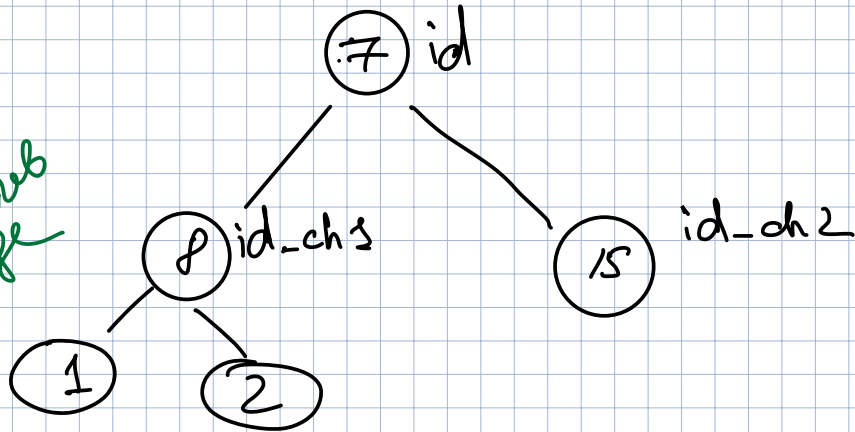
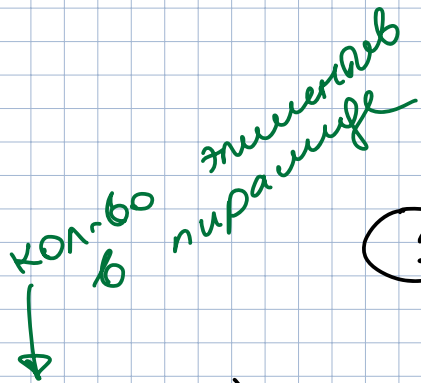
Асимптотика:

$$T(N) = \underline{O}(n)$$

$$\mu(n) = \underline{0}(1)$$

Плохой случай 2

Sift Down



SiftDown(arr, N, id)

```
while ( id < N ) {
```

$$\text{id}_{ch_1} = \text{id} \otimes 2 \otimes 1$$

$$\vec{id_ch2} = id * 2 + 2$$

$$\text{id_max} = \text{id}$$

// B tee monomul
naxennyu
in pl.

if $(id_ch \leq N)$ and $(arr[id_ch]^{upper} > arr[id_max])$

```

    ↳ id_max = id_ch2
    if (id_ch2 < N) and (arr[id_ch2] > arr[id_max])
        ↳ id_max = id_ch2

    if (id != id_max)
        ↳ SWAP(arr[id], arr[id_max])
        id = id_max
    else
        ↳ break
}

```

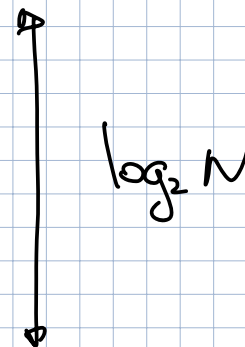
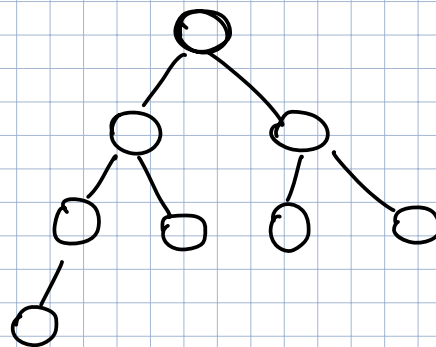
Асимптотика: $T(N) = O(N)$ $M(N) = O(1)$

III Оценка высоты пирамиды

$$2^x = N/2$$

$$2^{x+1} = N$$

$$x+1 = \log_2 N$$



$$x = \log_2 N - 1$$

IV

extract Max(arr, N)

→ swap(arr[0], arr[N-1])

N--;

max = arr[N-1]

sift Down(arr, 0)

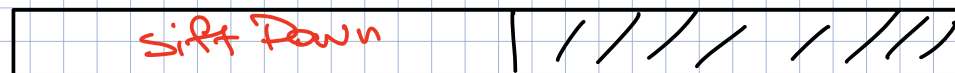
Асимптотика: $O(N) = O(\log N)$

V

Быстро построить кучу

→ Есть массив (N элементов)

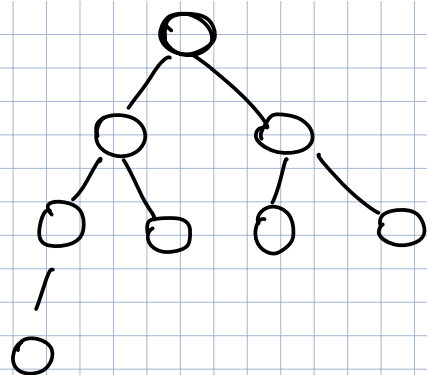
→ За $O(N)$ постро. пирамиды



sift Down

для всех
этих
элементов.

↑ последний элемент



$h=2$

$h=1$

$h=0$

Доказательство

асимптотики:

$$H = \log_2 N$$

$$\leq \frac{N}{2^{h+1}}$$

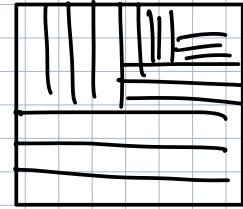
— кол-во элементов на уровне h

Время работы:

$$\begin{aligned}
 T(N) &= \sum_{h=0}^{\log_2 N} \frac{N}{2^{h+1}} O(h) = \\
 &= \frac{N}{2} \left(\sum_{h=0}^{\log_2 N} \frac{O(h)}{2^h} \right) = \\
 &= O \left(\frac{N}{2} \cdot \underbrace{\sum_{h=0}^{\log_2 N} \frac{h}{2^h}}_2 \right) = \\
 &= O \left(\frac{N}{2} \cdot 2 \right) = O(N)
 \end{aligned}$$

Примерами:

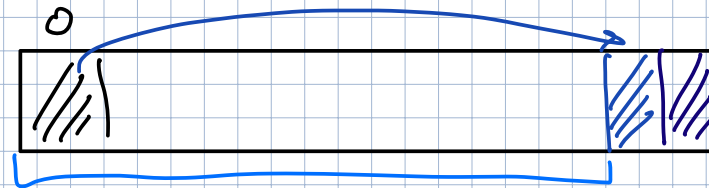
$$\begin{aligned}
 \sum_{x=0}^{\infty} \frac{x}{2^x} &= 0 + \frac{1}{2} + \frac{2}{4} + \sum_{x=3}^{\infty} \frac{x}{2^x} = \\
 &= 1 + \sum_{x=3}^{\infty} \frac{x}{2^x} \leq 2
 \end{aligned}$$



(VI)

Heap Sort (пирамидальная сортировка)

Алго: \rightarrow extract Max $\times N$ раз



корректная
пирамида

доступные
элементы

$IP(A) = \frac{1}{N}$
 $O(1)$

```

push (arr, N, capacity, elem)
{
    if (N >= capacity - 1)
        O(N) resize(arr, N, elem)

```

```

    arr[N] = elem;
    N++;
    siftUp(arr, N)

```

```

}

```

```

resize((int*) arr, N, int capacity)
{

```

```

    int* new_arr = new int [2*capacity];

```

```

    for i = 0 ... capacity
        new_arr[i] = arr[i]

```

```

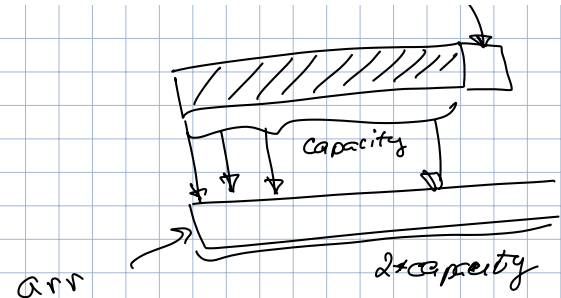
    delete [] arr;

```

```

    arr = new_arr
    capacity = capacity * 2
}

```



O(N)

