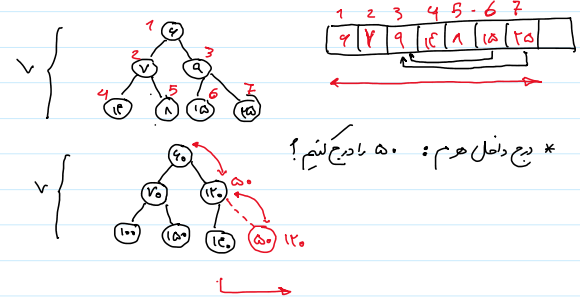


موضوع: ادامه صف اولویت + مقدماتی از درخت سازی *

صف اولویت ← امکان حذف x
 \leftarrow حذف مقدار x
 \leftarrow delete min

← BST با کلاس:

Heap: 1. جدولی مثل
 2. خاصیت heap



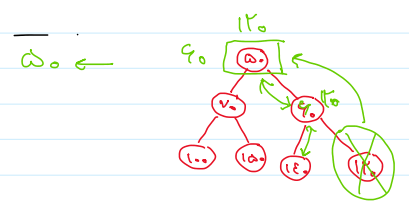
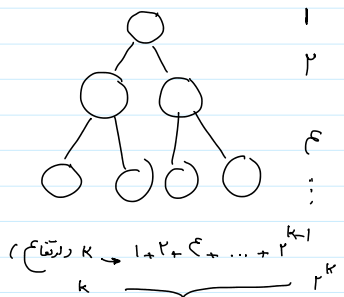
* درج داخل هرم: ۵۰ را درج کنیم!



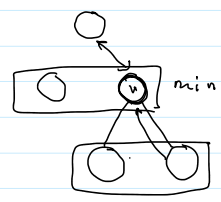
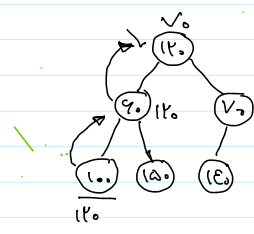
درج: عددی که می‌خواهیم درج کنیم را در راس مشخص قرار می‌دهیم و سپس هر بار باید بر مقایسه و در صورت لزوم swap می‌کنیم

bubble up

order: $O(\log n)$



حذف کنیم:
 delete min:
 کمینه همیشه ریشه



bubble down

$O(\log n)$

Heapsort: مرتب‌سازی هری

ورودی: A با اندازه n
 خروجی: A به صورت مرتب‌شده

الگوریتم:

* 1. A را تبدیل به هرم کنیم ← ؟

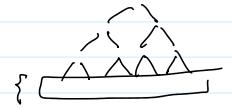
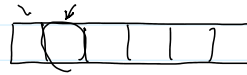
۲- n بار عمل delete min را انجام دهیم $\leftarrow O(n \log n)$

تبدیل آرایه به هم:



۱- عنصر را یکی یکی به Heap اضافه کنیم:
for $(i: 2 \rightarrow n)$
bubble up(i)

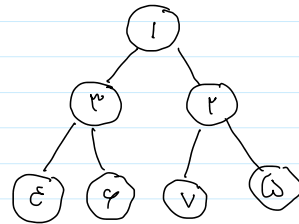
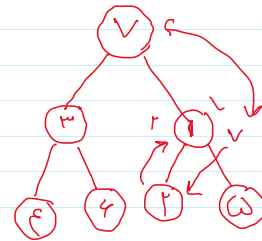
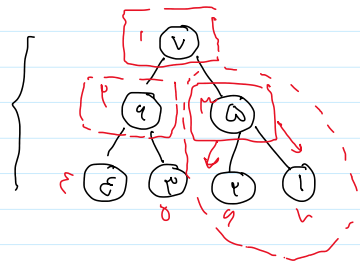
$O(n \log n)$



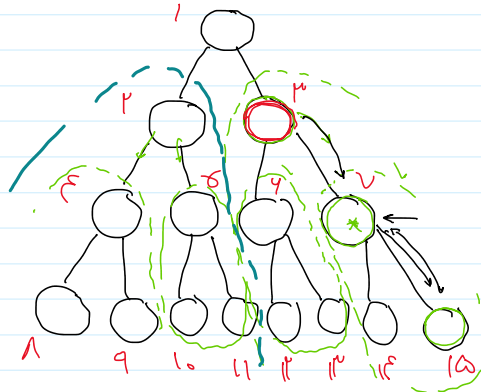
۲- for $(i: \lfloor \frac{n}{2} \rfloor \rightarrow 1)$

bubbleDown(i)

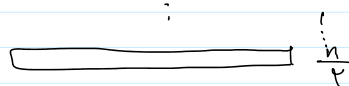
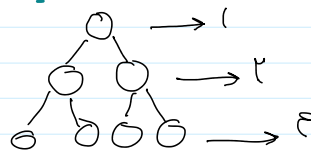
$n=7 \quad \lfloor \frac{n}{2} \rfloor = 3$



باید جست کنیم bubbleDown هر عنصر چند swap نیاز دارد.



۰ : ۱۵-۸
۱ : ۸-۷
۲ : ۷-۴
۳ : ۴-۱



$\frac{n}{2} : 0$

$\frac{n}{4} : 1$

$\frac{n}{8} : 2$

\vdots

$1 : \log n$

$$\frac{n}{2} \times 0 + \frac{n}{4} \times 1 + \frac{n}{8} \times 2 + \dots + 1 \times \log n$$

$$\begin{aligned}
 & \frac{n}{2} \times 0 + \frac{n}{\varepsilon} \times 1 + \frac{n}{\lambda} \times 2 + \dots + 1 \times \log n \\
 &= n \left[\frac{1}{\varepsilon} + \frac{2}{\lambda} + \frac{4}{16} + \frac{8}{32} + \dots \right] \\
 & \quad \frac{1}{2} = \frac{1}{\varepsilon} \quad \frac{1}{\lambda} \quad \frac{1}{16} \quad \frac{1}{32} \dots \\
 & n \left[\frac{1}{2} + \left(\frac{1}{\lambda} + \frac{2}{16} + \frac{4}{32} + \dots \right) \right] \\
 & \quad \frac{1}{4} \quad \frac{1}{16} \\
 & n \left[\frac{1}{2} + \frac{1}{\varepsilon} + \left(\frac{1}{16} + \frac{2}{32} + \dots \right) \right] \\
 & n \left[\frac{1}{2} + \frac{1}{\varepsilon} + \frac{1}{\lambda} + \frac{1}{16} + \dots \right] \ll n
 \end{aligned}$$

Heap

موضوع: Map و جدول درهم سازی.

آشنایی با Map ADT و Dictionary.

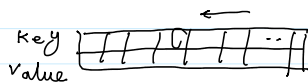
Dictionary (map): هدف ذخیره تعدادی دوتایی به صورت (key, value) است.

مثال: دیکشنری:

عملیات: insert(key, value)

← search(key) کیدها یاب

delete(key)



۱- آرایه نامرتب

insert: $O(1)$

search: $O(n)$

delete: $O(n)$



۲- آرایه مرتب

(key, value)

insert: $O(n)$

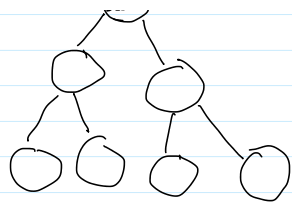
search: $O(\log n)$

delete: $O(n)$

۳- با استفاده از درج باینری.

insert: $O(\log n)$





۱- جستار به جستار

insert : $O(\log n)$

delet : $O(\log n)$

search : $O(\log n)$

Q(1) ؟

مثال : فرض کنید ی دانم که همه کلیدها در بازه $\{1, 2, \dots, 9999\}$ و $\{0\}$ است.

(۲۵, masoud)

0	1	2	25	9999
-1	-1	-1	-1	-1

* اگر مجموعه کلیدها به اندازه کافی کوچک باشند ،

ما رقم آخر شماره دانشجویی ها :