

# طراحی الگوریتم

جلسه ششم

---

- فصل ۶:

- مرتب سازی درجا (inplace): اگر میزان حافظه کمکی وابسته به اندازه ورودی نباشد.

- $\left\{ \begin{array}{l} \textit{Quic Sort} \\ \textit{Insertion Sort} \\ \textit{Selection Sort} \\ \textit{heap Sort} \end{array} \right.$

# Heap

- تعریف Heap:
- Max – heap: درخت دودویی کامل که مقدار هر گره بزرگتر مساوی از مقدار فرزندان.
- Min – heap: درخت دودویی کامل که مقدار هر گره کوچکتر مساوی از مقدار فرزندان.
- ذخیره یک Heap در آرایه:

array:	0	1	2	3	4	5	6	7
		A	B	C	D	E	F	

---

*Parent* (*i*)  $\rightarrow \lfloor i/2 \rfloor$

*Lchild* (*i*)  $\rightarrow 2 * i \leq n$

*Rchild* (*i*)  $\rightarrow 2i + 1 \leq n$

*Size of heap*  $\rightarrow \text{current} - \text{size}[A]$

*Maximom size of heap*  $\rightarrow \text{Max} - \text{sice}[A]$

*Max - Heap if y* (*A, i*) {

*index* = *i*;

*if* ( $(2i \leq \text{curret} - \text{size}[A])$

$\&\&(A[2i] > A[\text{index}]))$

*index* =  $2i$ ;

*if* ( $(2i + 1 \leq \text{current} - \text{size}[A]) \&\&(A[2i + 1] > A[\text{index}]))$  *index* =  $2i + 1$ ;

*if* (*index*  $\neq i$ ) {

$A[i] \leftrightarrow A[\text{index}]$ :

*Max - Heapify* (*A, index*);\*

}

}

---

$d = \text{عمق درخت}$

$n = \text{تعداد گره های درخت} = 2^0 + 2^1 + 2^2 \dots + 2^{d-1} + 2^d / 2$

$$= 2^0 + 2^1 + \dots + 2^{d-1} + 2^{d-1} = \frac{3}{2} 2^d - 1$$

$n' = \text{تعداد بیشترین گره ها در زیر درخت سمت چپ}$

$$= 2^1 / 2 + 2^2 / 2 + \dots + 2^{d-1} / 2 + 2^d / 2$$

$$= 2^0 + 2^1 + \dots + 2^{d-1} = 2^d - 1$$

$$\frac{n'}{n} = \frac{2^{d-1} - 1}{\frac{3}{2} \times 2^d - 1} \simeq 2/3$$

برای مقایسه

$$T(n) \leq T(2n/3) + \overset{\uparrow}{\theta(1)} \xrightarrow{\text{حالت دوم قضیه اصلی}} T(n) = \theta(\log n)$$

\*  $T(2n/3)$ : چون اندازه هر یک از درخت های  $i$  در بدترین حالت با توجه به اینکه درخت

کامل است در زمانی رخ می دهد که آخرین سطر درخت نیمه پر باشد (در بدترین حالت

بیشترین گره های موجود در زیر درخت ها  $2n/3$  است).

نکته: در نمایش آرایه برای مرتب سازی هیپ  $n$  عنصری، برگ ها، گره هایی با اندلیس های

$[n/2] + 1, [n/2] + 2, \dots, n$  هستند.

آرایه  $A$  را تبدیل به Max – Heap می کند.



