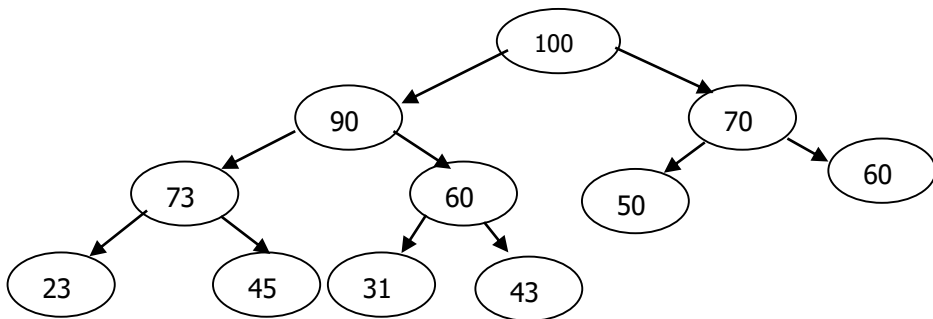


Thuật toán sắp xếp vun đống (Heap sort)

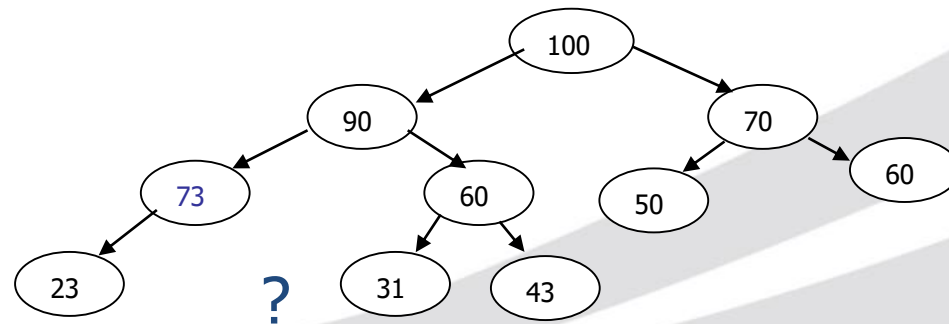
Cây Heap



- **Cây heap (đồng)** là một cây nhị phân được sắp xếp theo khóa của các nút với các tính chất sau:
 - Giá trị khóa của nút gốc \geq giá trị khóa của hai con
 - Tất cả các mức đều đầy trừ mức thấp nhất có thể thiếu một số nút
 - Các nút lá phải xuất hiện liên tiếp từ trái qua phải
- Như vậy nút gốc có giá trị khóa lớn nhất
- Ví dụ:



Cây Heap



Không phải cây Heap

Mảng biểu diễn cây heap



- Mảng $A[1], \dots, A[n]$ là mảng biểu diễn cây heap nếu:
 - $A[i] \geq A[2i]$ và $A[i] \geq A[2i+1]$ với $i=1..n/2$
- Như vậy phần tử đầu của mảng có giá trị lớn nhất

• Ví dụ: A

100	90	70	73	60	50	60	23	45	31	43
-----	----	----	----	----	----	----	----	----	----	----

$$A[1] \geq A[2], A[1] \geq A[3]$$

$$A[3] \geq A[6], A[3] \geq A[7]$$

$$A[2] \geq A[4], A[2] \geq A[5]$$

$$A[4] \geq A[8], A[4] \geq A[9]$$

$$A[5] \geq A[10], A[5] \geq A[11]$$

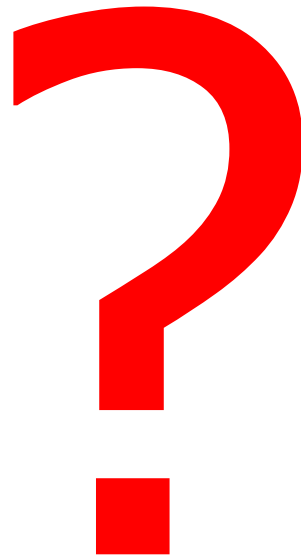
Thuật toán sắp xếp vun đống



Ý tưởng:

- Tạo mảng $A[1], \dots, A[n]$ biểu diễn cây Heap.
- Tráo đổi phần tử $A[1]$ với phần tử $A[n]$.
- Tạo mảng $A[1], \dots, A[n-1]$ biểu diễn cây heap
- Tráo đổi phần tử $A[1]$ với phần tử $A[n-1]$.
- Lặp lại quá trình trên đến khi mảng chỉ còn 1 phần tử

Tạo đồng



Tạo mảng biểu diễn cây heap



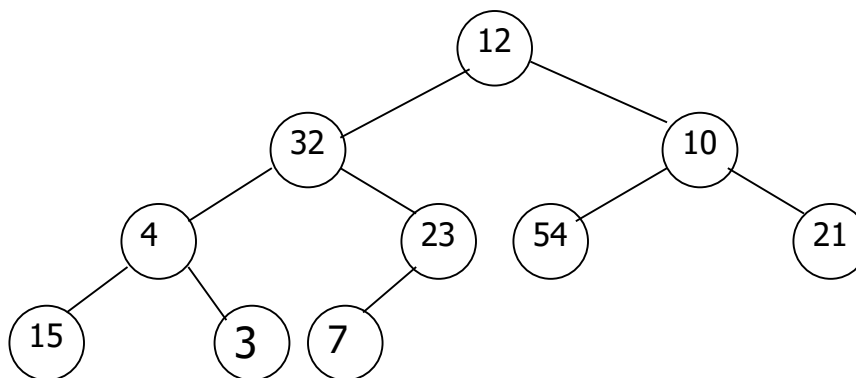
- Theo tính chất của mảng biểu diễn cây Heap thì các phần tử từ $n/2+1$ đến n không cần điều kiện ràng buộc. Vì vậy ta thực coi các phần tử này đã thỏa mãn điều kiện cây heap.
- Ta thực hiện:
 - Bổ sung phần tử $n/2$ vào $A[n/2+1], \dots, A[n]$ để được mảng gồm $A[n/2], \dots, A[n]$ thỏa mãn kiện
 - Bổ sung phần tử $n/2-1$ vào $A[n/2], \dots, A[n]$ để được mảng gồm $A[n/2-1], \dots, A[n]$ thỏa mãn kiện
 - Và cứ tiếp tục làm như vậy cho đến khi bổ sung phần tử $A[1]$ vào $A[2], \dots, A[n]$ để được mảng gồm $A[1], \dots, A[n]$ thỏa mãn điều kiện

Ví dụ



Cho mảng như dưới đây, hãy biến đổi mảng để được mảng thỏa mãn tính chất mảng biểu diễn cây heap

12	32	10	4	23	54	21	15	3	7
----	----	----	---	----	----	----	----	---	---



Cây tương ứng với mảng

Mô tả trên mảng: $N=10$



12	32	10	4	23	54	21	15	3	7
----	----	----	---	----	----	----	----	---	---

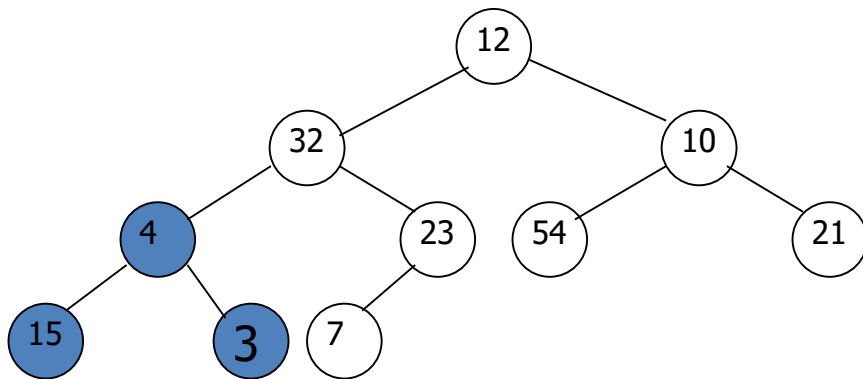
$i=5$

12	32	10	4	23	54	21	15	3	7
----	----	----	---	----	----	----	----	---	---

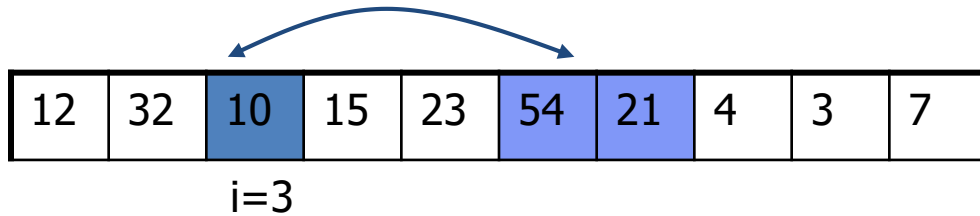
$i=4$

- Đổi chỗ $A[5]$ và $A[10]$ nếu $A[5] < A[10]$

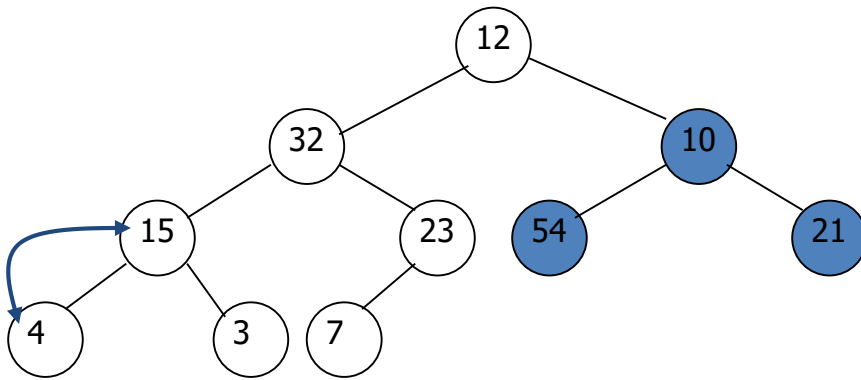
- Tính $\max(A[8], A[9])$. Nếu $A[4] < \max$ thì đổi chỗ $A[4]$ với phần tử đạt max



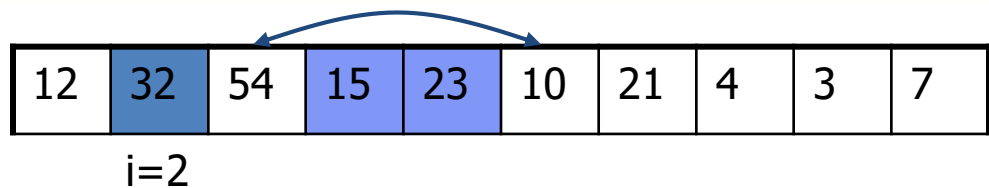
Cây tương ứng với mảng



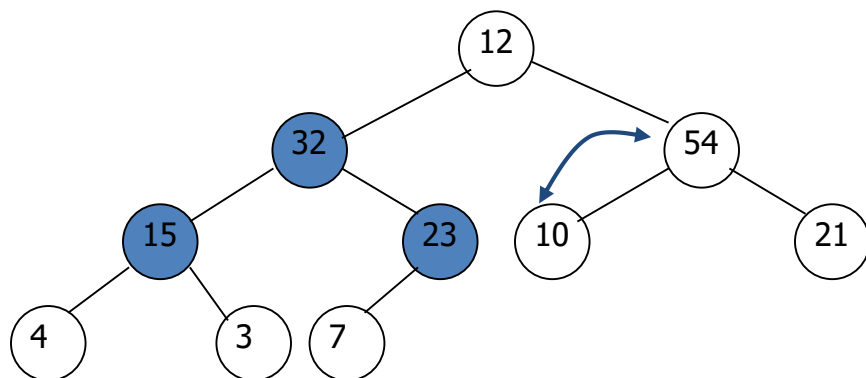
- Tính $\max(A[6], A[7])$. Nếu $A[3] < \max$ thì đổi chỗ $A[3]$ với phần tử đạt max



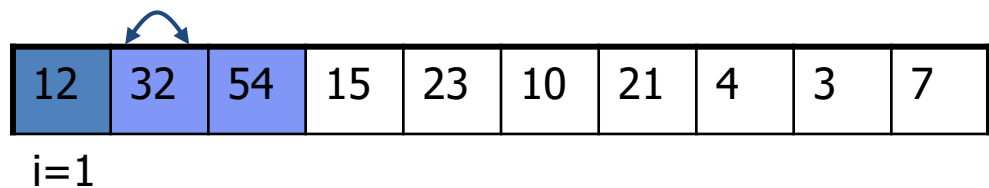
Cây tương ứng với mảng



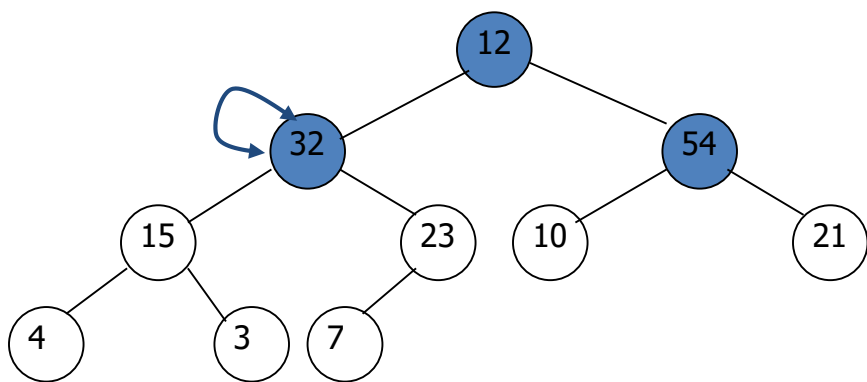
- Tính $\max(A[4], A[5])$. Nếu $A[2] < \max$ thì đổi chỗ $A[2]$ với phần tử đạt max



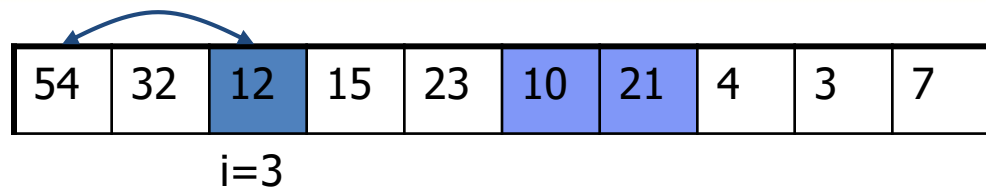
Cây tương ứng với mảng



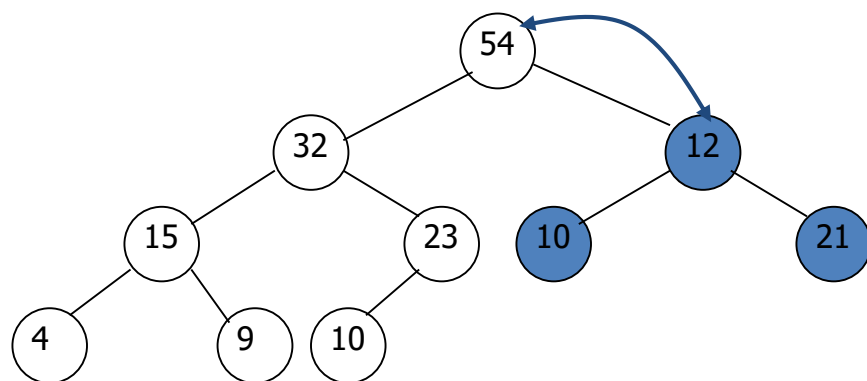
- Tính $\max(A[2], A[3])$. Nếu $A[1] < \max$ thì đổi chỗ $A[1]$ với phần tử đạt max



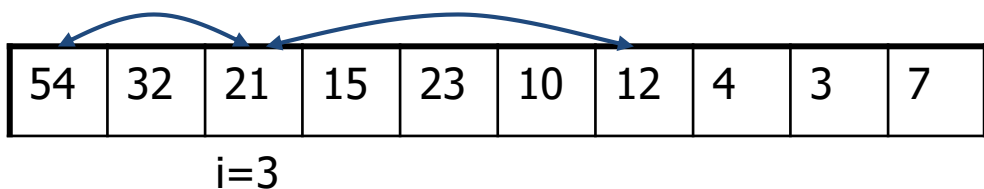
Cây tương ứng với mảng



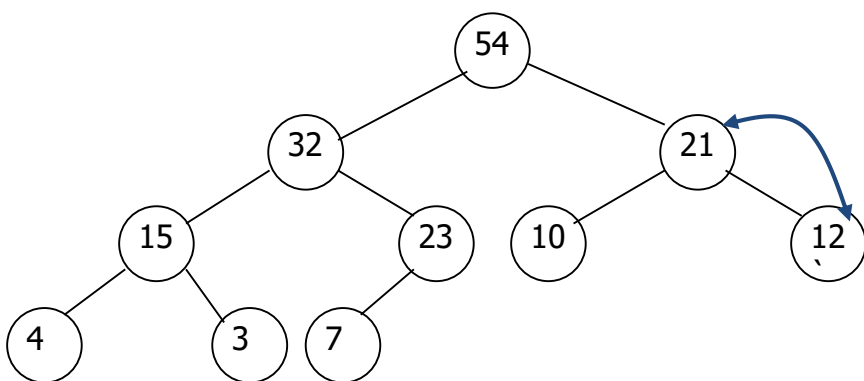
- Tính $\max(A[6], A[7])$. Nếu $A[3] < \max$ thì đổi chỗ $A[3]$ với phần tử đạt max



Cây tương ứng với mảng



- Tính $\max(A[6], A[7])$. Nếu $A[3] < \max$ thì đổi chỗ $A[3]$ với phần tử đạt max



Cây heap

Thuật toán bổ sung một phần tử để tạo mảng biểu diễn cây heap



Algorithm *Pushdown* (Array A , i , n);

Input: số nguyên i , n , mảng $A[i], \dots, A[n]$, trong đó $A[i+1], \dots, A[n]$ thỏa mãn tính chất cây heap

Output: Mảng $A[i], \dots, A[n]$ thỏa mãn tính chất cây heap

$j \leftarrow i$; $kt \leftarrow 0$;

while ($j \leq n/2$) and ($kt=0$) **do**

if $2*j = n$ **then**

$max \leftarrow 2*j$;

else

if $A[2*j].key \leq A[2*j+1].key$ **then**

$max \leftarrow 2*j+1$

else

$max \leftarrow 2*j$;

if $A[j].key < A[max].key$ **then**

 swap ($A[j]$, $A[max]$);

$j \leftarrow max$;

else

$kt \leftarrow 1$;

Thuật toán sắp xếp vun đống



Algorithm *Heapsort*(Array A , n);

Input: Mảng A có n phần tử và số nguyên n

Output: Mảng A được sắp theo thứ tự tăng dần của thuộc tính khóa

for $i \leftarrow n/2$ downto 1 do

 Pushdown(A , i , n);

for $i \leftarrow n$ downto 2 do

 swap($A[1]$, $A[i]$);

 Pushdown(A , 1, $i-1$);

Ví dụ:



Mô tả quá trình sắp xếp của dãy số

12 43 11 34 23 43 12 435

Thời gian chạy



- Thời gian thực hiện thủ tục Pushdown.
 - Là t/g thực hiện của vòng lặp while.
 - Gọi k là số lần lặp, ta có $i \cdot 2^k \leq n$ hay $k \leq \log_2(n/i)$.
 - T/g thực hiện hàm Pushdown (A,i, n) là $O(\log(n/i))$
- Xét thủ tục HeapSort
 - Vòng lặp for đầu có số lần lặp là $n/2$
 - Mỗi lần gọi hàm Pushdown 1 lần. Do đó t/g thực hiện là $O(\log_2 n)$.
 - Tương tự, vòng lặp for thứ 2 có số lần lặp là $n-1$. $O(n \log_2 n)$.
 - Vì vậy t/g thực hiện HeapSort là $O(n \log_2 n)$.

Hết