

COMP 251

程序代写代做 CS编程辅导

Algorithms



Structures (Winter 2022)

Heaps
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Outline

程序代写代做 CS编程辅导

- Introduction.
- Operations.
- Application.



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Introduction – Heap data structure

程序代写代做 CS 编程辅导

- Tree-based data structure (here, binary tree, but we can also use k-ary trees)



- Max-Heap

- Largest element is stored at the root.
- for all nodes i , excluding the root, $A[\text{PARENT}(i)] \geq A[i]$.

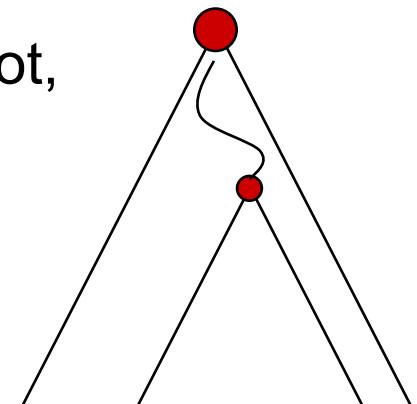
- Min-Heap

Assignment Project Exam Help

- Smallest element is stored at the root.
- for all nodes i , excluding the root, $A[\text{PARENT}(i)] \leq A[i]$.

<https://tutorcs.com>

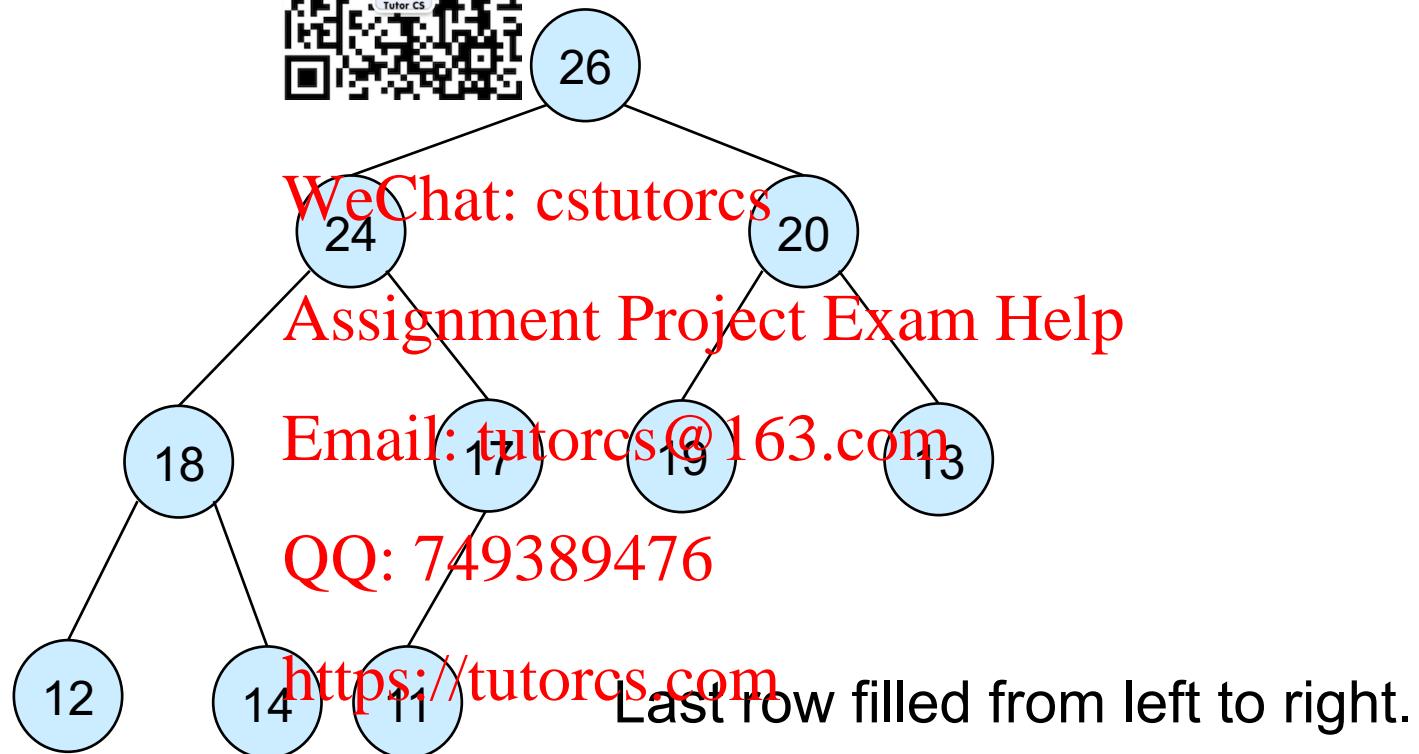
- Tree is filled top-down from left to right.



Introduction – Heap - example

程序代写代做 CS 编程辅导

Max-heap as a binary tree

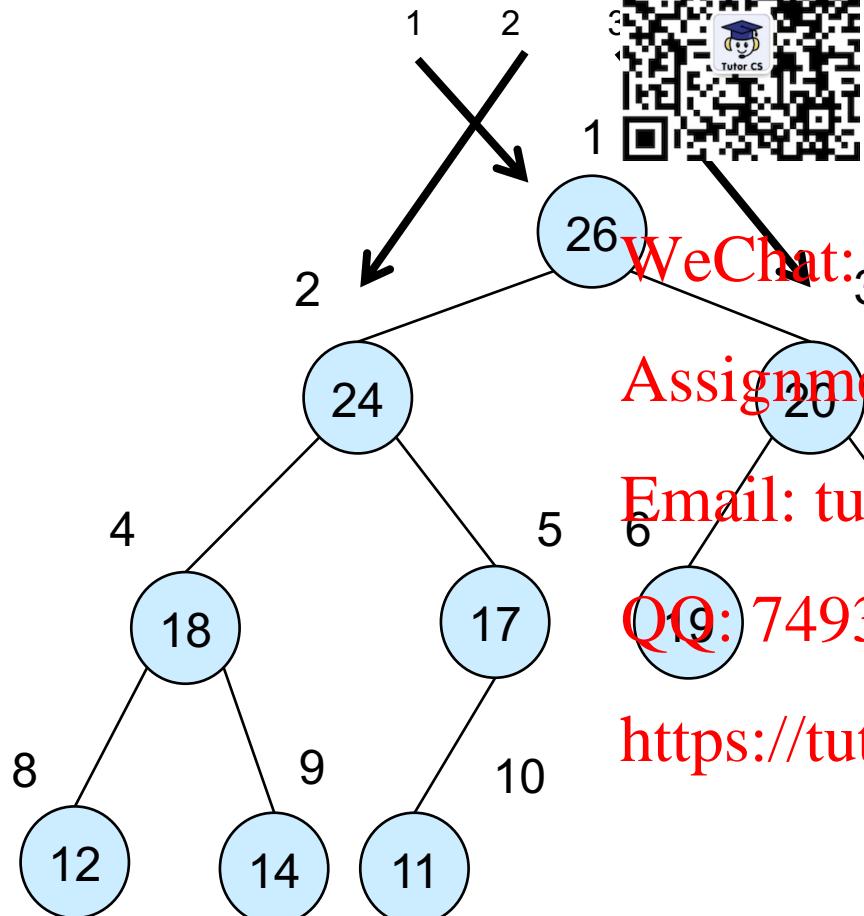


Introduction – Heap as arrays - example

程序代写代做 CS 编程辅导



Max-heap as an array.



Map from array elements to tree nodes:

- Root : $A[1]$
- Left[i] : $A[2i]$
- Right[i] : $A[2i+1]$
- Parent[i] : $A[\lfloor i/2 \rfloor]$

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Introduction – Heap - Height

程序代写代做 CS 编程辅导

- *Height of a node in a tree:* the number of edges on the longest simple path from the node to a leaf.
- *Height of a heap = height of the root* = $O(\lg n)$.
- Most Basic operations on a heap run in $O(\lg n)$ time
- Shape of a heap



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

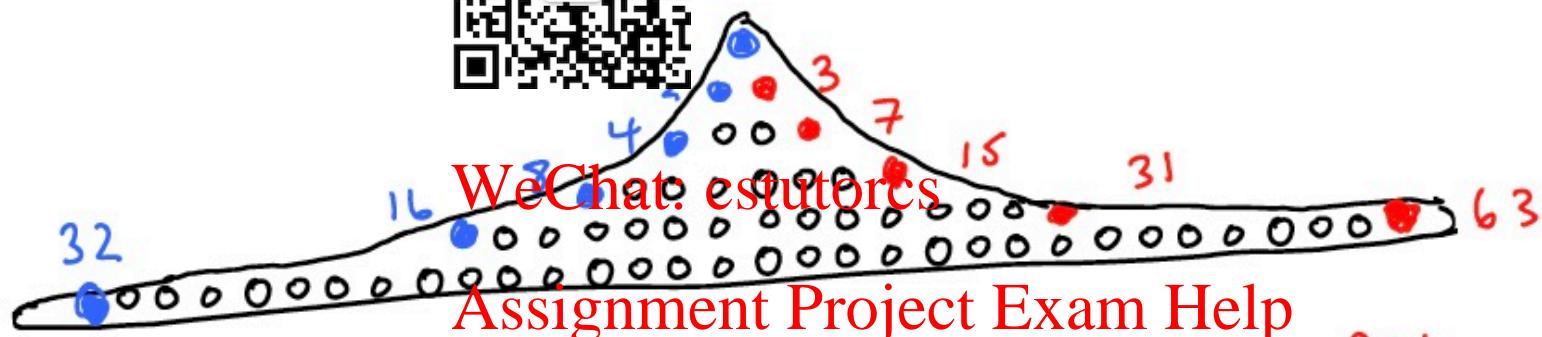
QQ: 749389476

<https://tutorcs.com>

Introduction – Heap - Height

程序代写代做 CS 编程辅导

Consider a complete binary tree of height h , all levels full.



Assignment Project Exam Help

level l has 2^l nodes: $2^0, \dots, 2^{l+1} - 1$

Email: tutorcs@163.com
QQ: 749389476

number of nodes: $n = 2^{h+1} - 1$

height of tree: $h = \log(n+1) - 1$

Taken from Langer2014

Outline

程序代写代做 CS编程辅导

- Introduction.
- Operations.
- Application.



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Operations – Maintaining the heap property

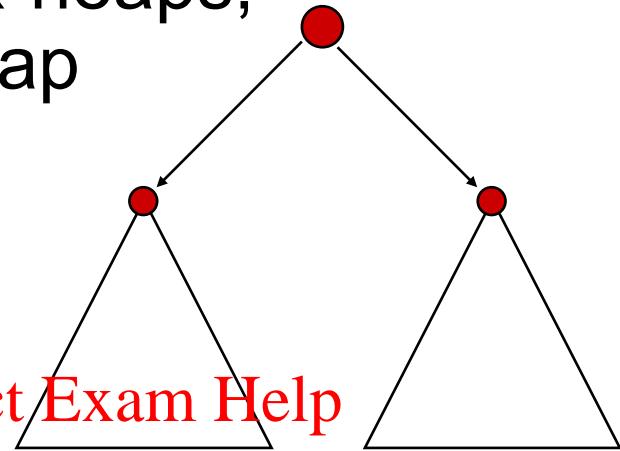
程序代写代做 CS 编程辅导

- Suppose two subtrees are max-heaps, but the root violates the max-heap property.



WeChat: cstutorcs

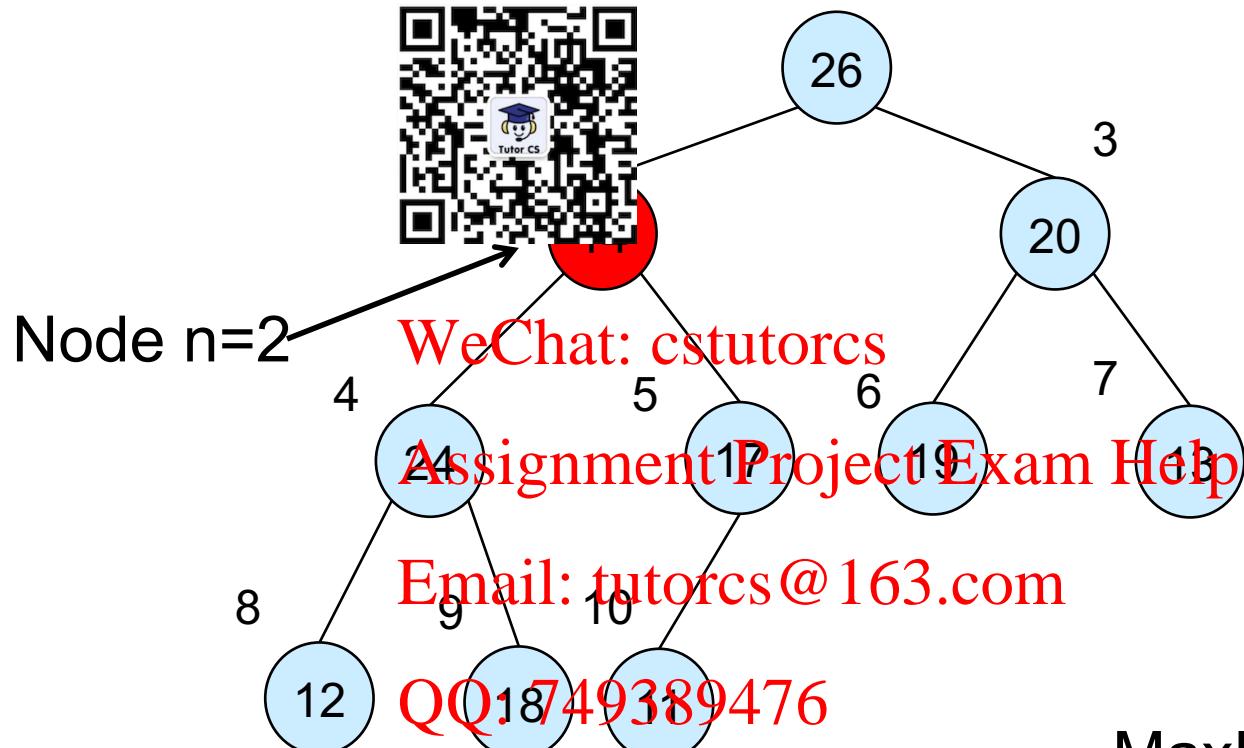
Assignment Project Exam Help



- Fix the offending node by exchanging the value at the node with the larger of the values at its children.
~~Email: tutorcs@163.com
QQ: 749389476~~
<https://tutorcs.com>
- The resulting tree may have a sub-tree that is not a heap.
- Recursively fix the children until all of them satisfy the max-heap property.

MaxHeapify – Example

程序代写代做 CS 编程辅导



Note: The call assumes that the left and right sub-trees of the node i are max-heaps, but that the node i might be smaller than its children (violating the max-heap property)

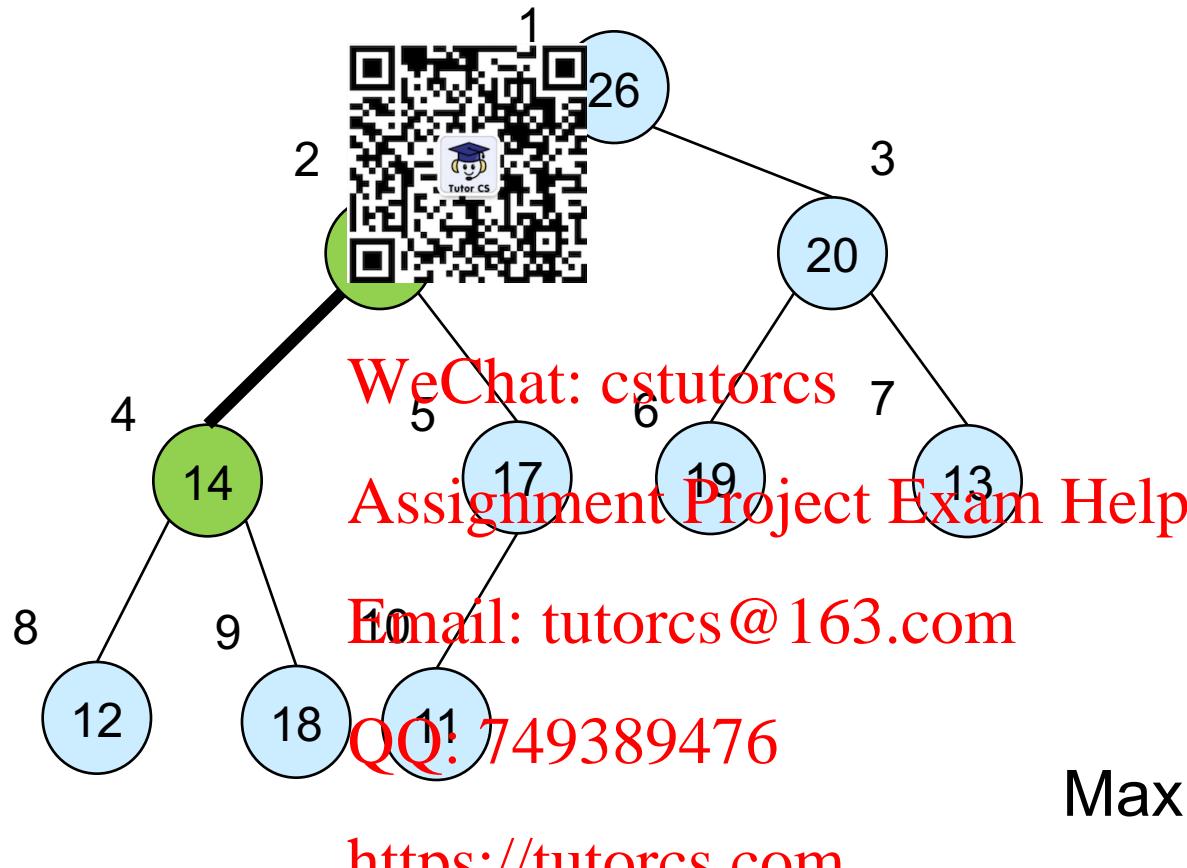
MaxHeapify(A , 2)

array

index

MaxHeapify – Example

程序代写代做 CS 编程辅导



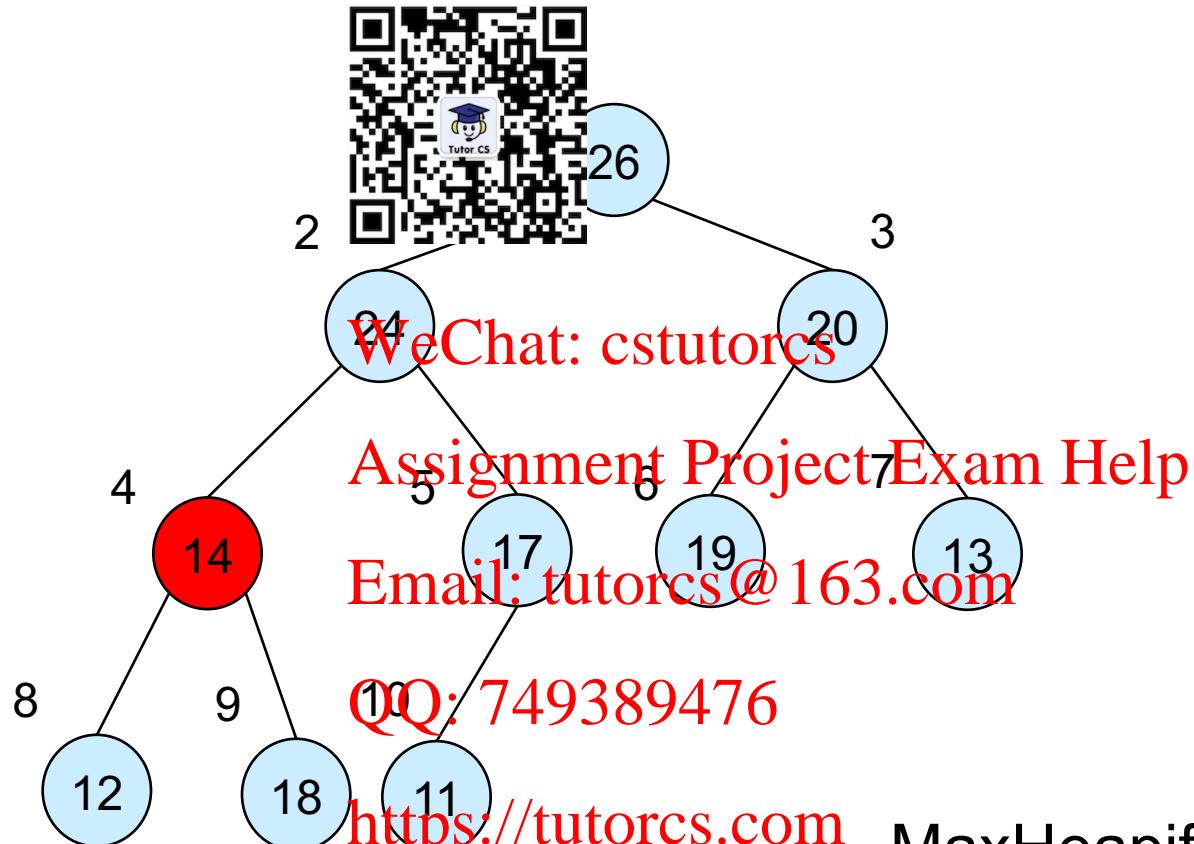
Note: The call assumes that the left and right sub-trees of the node i are max-heaps, but that the node i might be smaller than its children (violating the max-heap property)

MaxHeapify(A , 2)

array index

MaxHeapify – Example

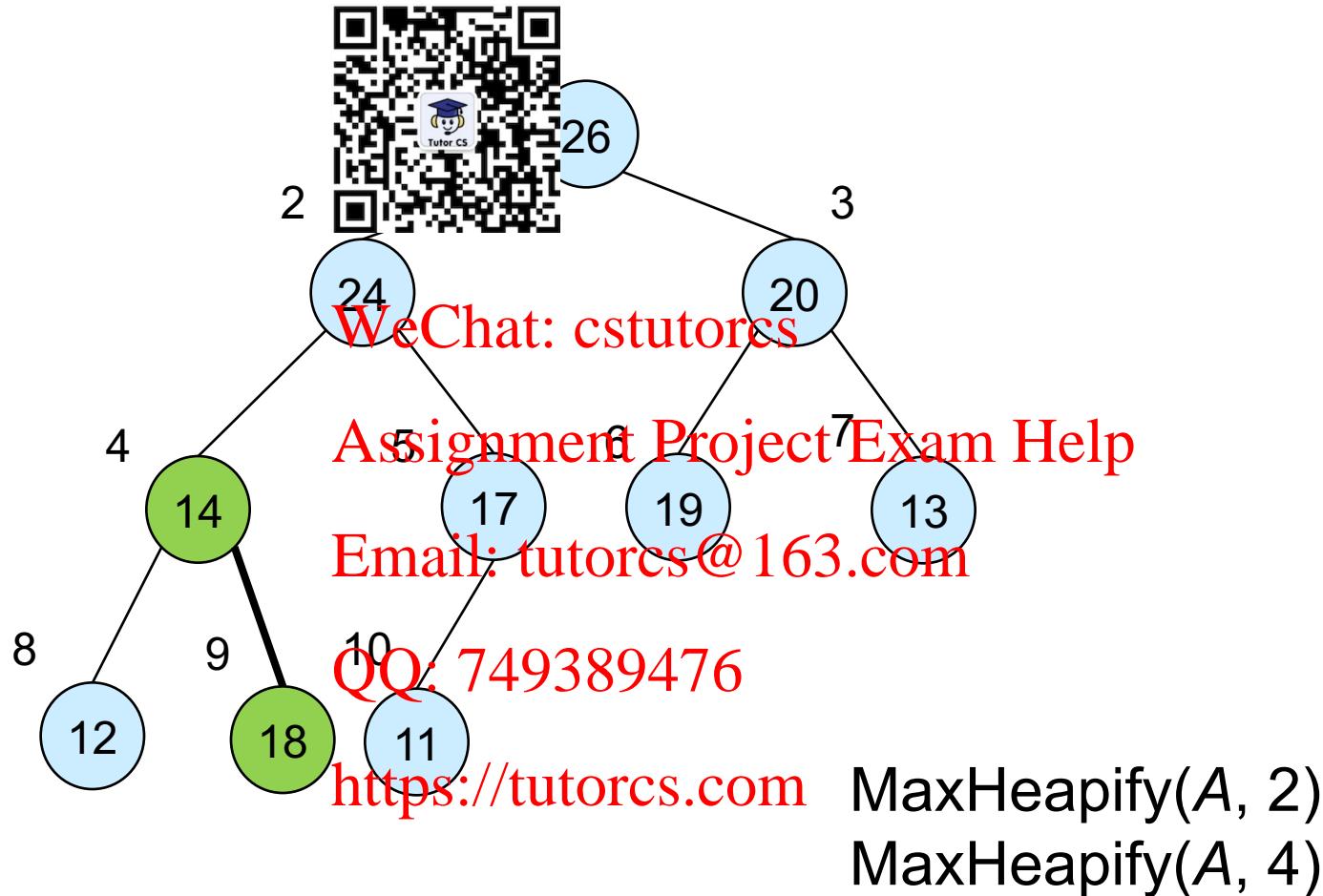
程序代写代做 CS 编程辅导



MaxHeapify(A , 2)
MaxHeapify(A , 4)

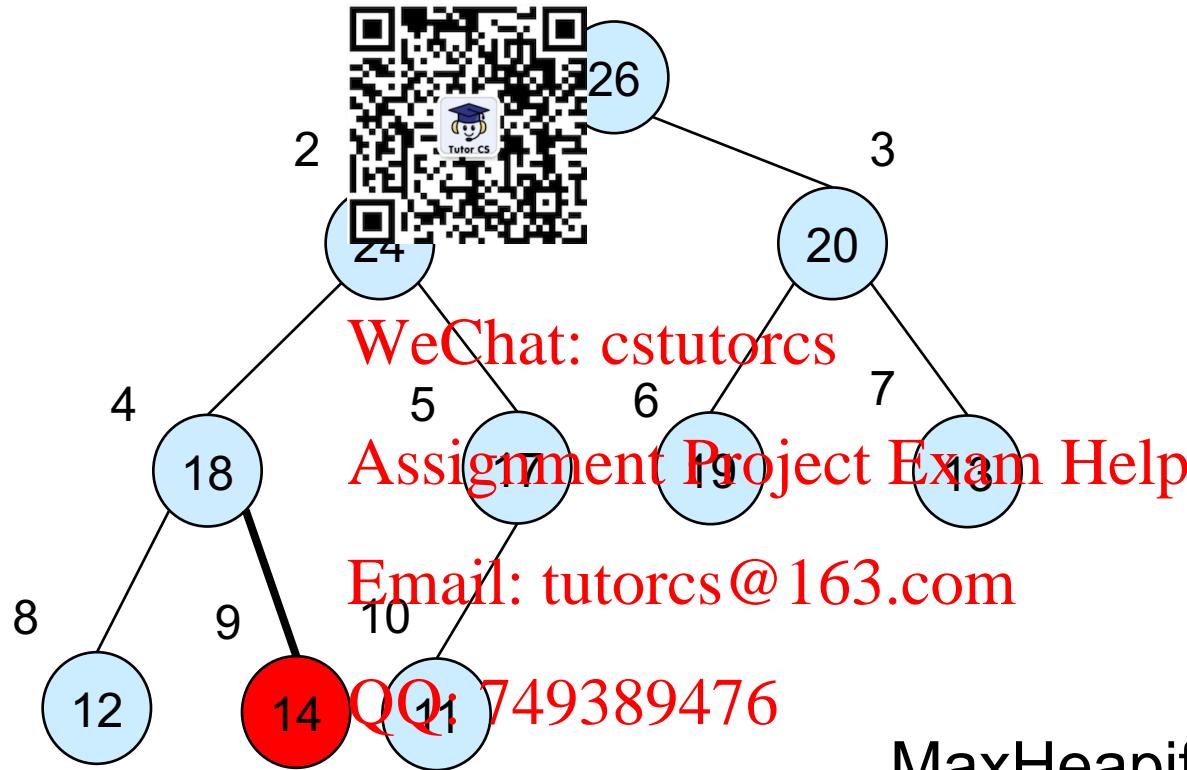
MaxHeapify – Example

程序代写代做 CS 编程辅导



MaxHeapify – Example

程序代写代做 CS 编程辅导



MaxHeapify(A , 2)
MaxHeapify(A , 4)
MaxHeapify(A , 9)

MaxHeapify – Example

程序代写代做 CS 编程辅导



MaxHeapify(A , 2)

MaxHeapify(A , 4)

MaxHeapify(A , 9)

MaxHeapify – Procedure

程序代写代做 CS 编程辅导

Assumption: Left(i) and Right(i) are max-heaps.

n is the size of the heap.

MaxHeapify(A, i)



1. $l \leftarrow \text{leftNode}(i)$
2. $r \leftarrow \text{rightNode}(i)$
3. $n \leftarrow \text{HeapSize}()$
4. **if** $l \leq n$ and $A[l] > A[i]$
then $\text{largest} \leftarrow l$
5. **else** $\text{largest} \leftarrow i$
6. **if** $r \leq n$ and $A[r] > A[\text{largest}]$
then $\text{largest} \leftarrow r$
7. **if** $\text{largest} \neq i$ <https://tutorcs.com>
8. **then** exchange $A[i] \leftrightarrow A[\text{largest}]$
9. **then** $\text{MaxHeapify}(A, \text{largest})$

WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
<https://tutorcs.com>

Time to determine if there is a conflict and find the largest children is $O(1)$

}

}

Time to fix the subtree rooted at one of i 's children is $O(\text{size of subtree})$

MaxHeapify – worst case

程序代写代做 CS 编程辅导

MaxHeapify(A, i)



- Size of a tree = n of nodes in this tree
- $T(n)$: time used for an input of size n
- $T(n) = T(\text{size of the largest subtree}) + O(1)$
- Size of the largest subtree $\leq 2n/3$ (worst case occurs when the last row Email: tutors@163.com is exactly half full)

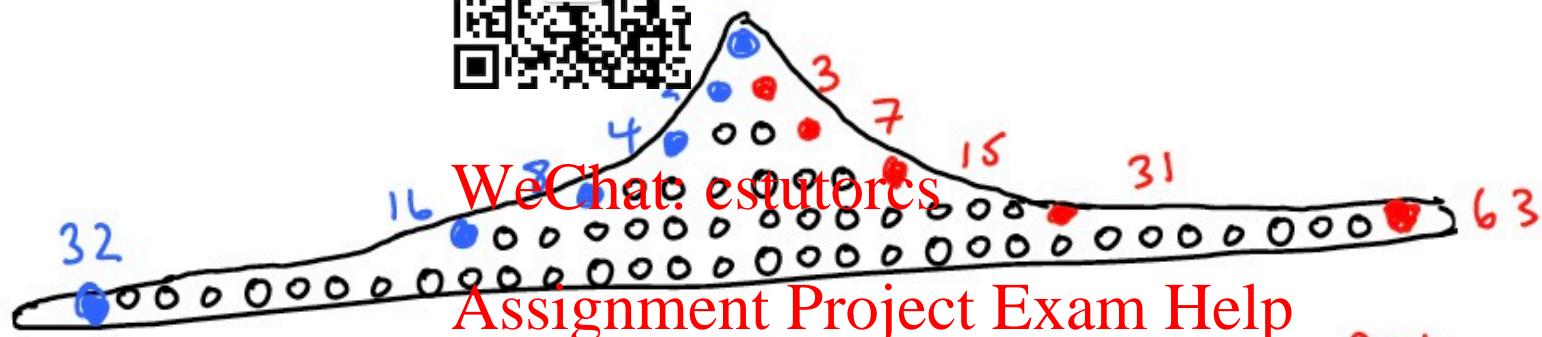
$$\Rightarrow T(n) \leq T(2n/3) + O(1) \xrightarrow{\text{Recurrence Relation}} T(n) = O(\lg n)$$

Alternately, MaxHeapify takes $O(h)$ where h is the height of the node where MaxHeapify is applied

Introduction – Heap - Height

程序代写代做 CS 编程辅导

Consider a complete binary tree of height h , all levels full.



Assignment Project Exam Help

level l has 2^l nodes: $2^0, \dots, 2^{l+1} - 1$

Email: tutorcs@163.com
QQ: 749389476

number of nodes: $n = 2^{h+1} - 1$

height of tree: $h = \log(n+1) - 1$

Taken from Langer2014

MaxHeapify – supplemental material

程序代写代做 CS 编程辅导

Taken from Langer2014

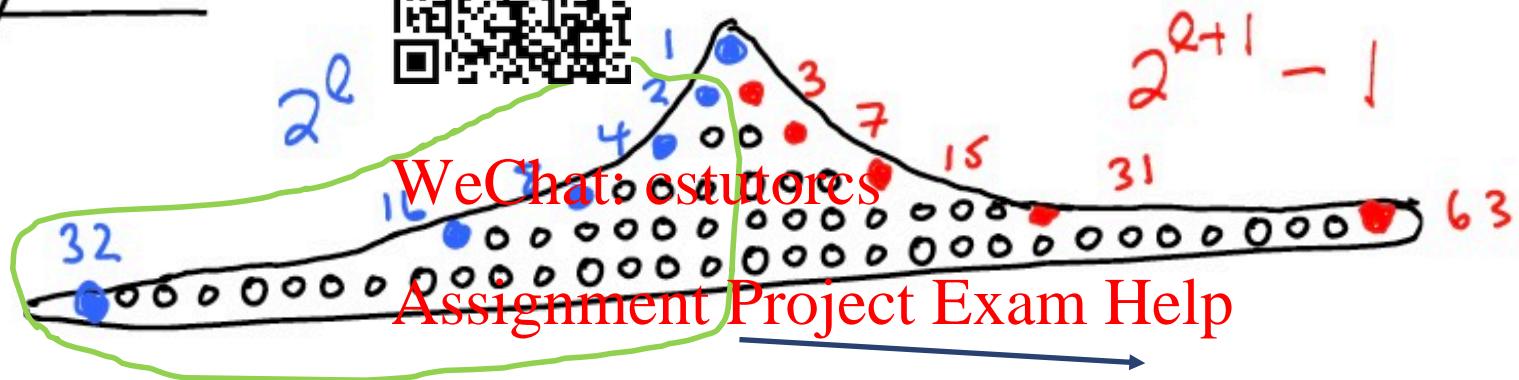
depth / level

0
1
2
3
4
5

2^0



e.g. tree height $h = 5$



$$\frac{(2^{h+1}-1)}{2} = \frac{2^{h+1}-1}{2}$$

QQ: 2749389476
https://tutorcs.com

root

MaxHeapify – supplemental material

程序代写代做 CS 编程辅导

Taken from Langer2014

depth / level

0
1
2
3
4
5

2^0



e.g. tree height $h = 5$

$2^{h+1} - 1$

63

WeChat: cstutorcs

Assignment Project Exam Help

of nodes in an added last row that is exactly half full.
Email: tutorcs@163.com

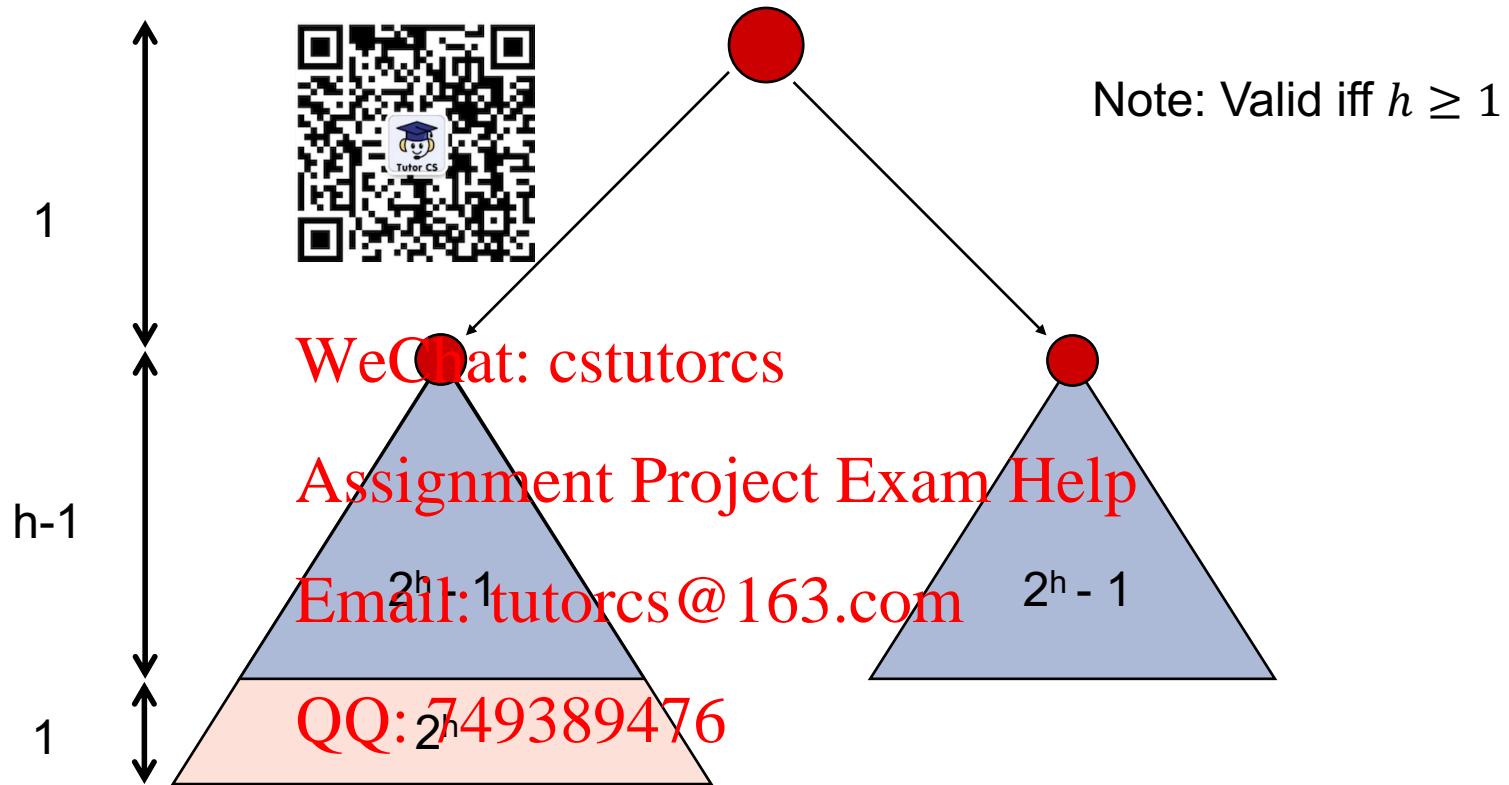
$$\frac{2^{h+1}}{2} = 2^h$$

QQ: 749389476

<https://tutorcs.com>

MaxHeapify – worst case

程序代写代做 CS 编程辅导

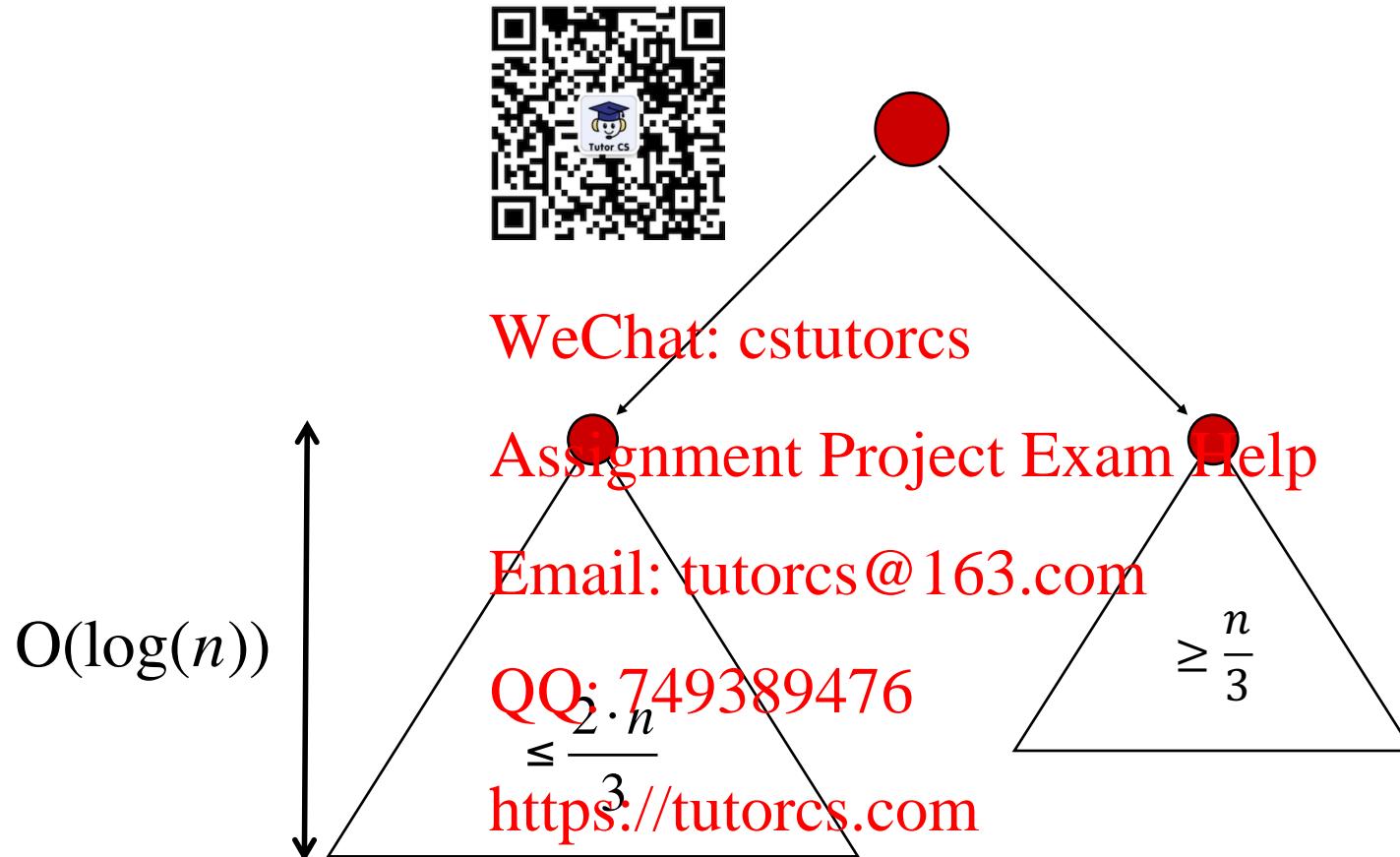


$$\text{Total in heap (n): } n = 2(2^h - 1) + 2^h + 1 = 3 \cdot 2^h - 1$$

$$\text{Total left subtree } n_{left} \leq 2^{h+1} - 1 = \frac{3}{3} \cdot 2 \cdot (2^h - \frac{1}{2}) = \frac{2}{3} \cdot (3 \cdot 2^h - \frac{3}{2}) \leq \frac{2}{3} \cdot n$$

MaxHeapify – worst case

程序代写代做 CS 编程辅导



Operations – Building a heap

程序代写代做 CS 编程辅导

- Use *MaxHeapify* to convert an array A into a max-heap.
- Call *MaxHeapify* on each element in a bottom-up manner.



BuildMaxHeap(A)

- WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
1. $n \leftarrow \text{length}[A]$
 2. **for** $i \leftarrow \lfloor n/2 \rfloor$ **downto** 1
 3. **do** ~~QQ~~ *MaxHeapify(A, i)*

<https://tutorcs.com>

Building a heap - Example

程序代写代做 CS 编程辅导

Input Array:

24	21	23	22	36	34	28	27
----	----	----	----	----	----	----	----



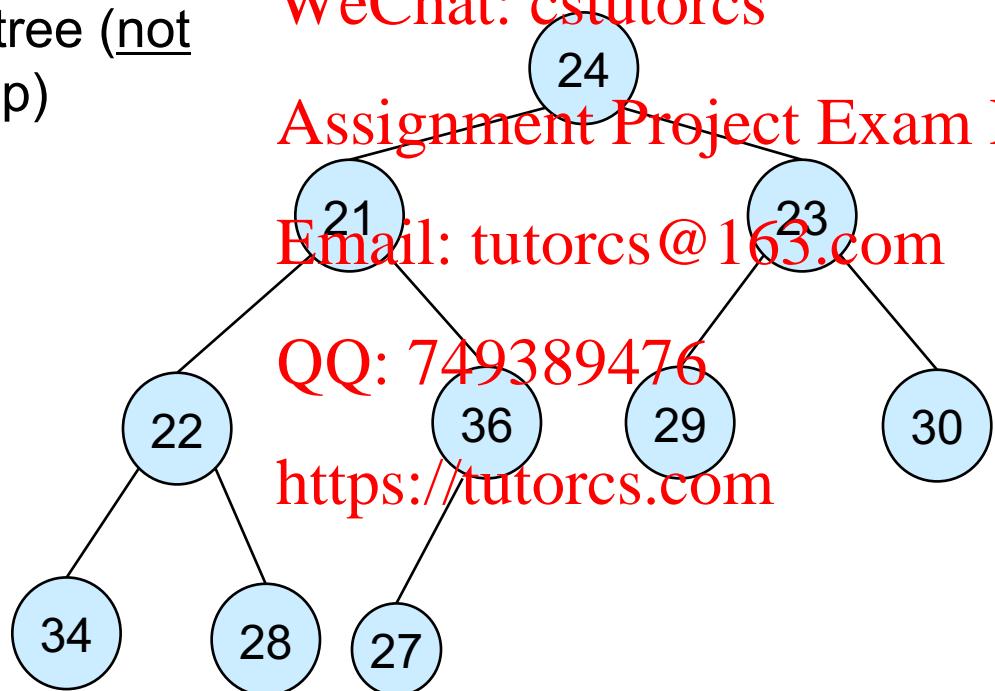
Starting tree (not
max-heap)

WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Building a heap - Example

程序代写代做 CS 编程辅导



Building a heap - Example

程序代写代做 CS 编程辅导



Building a heap - Example

程序代写代做 CS 编程辅导



Building a heap - Example

程序代写代做 CS 编程辅导



Building a heap - Example

程序代写代做 CS 编程辅导



Building a heap - Example

程序代写代做 CS 编程辅导



Building a heap - Example

程序代写代做 CS 编程辅导



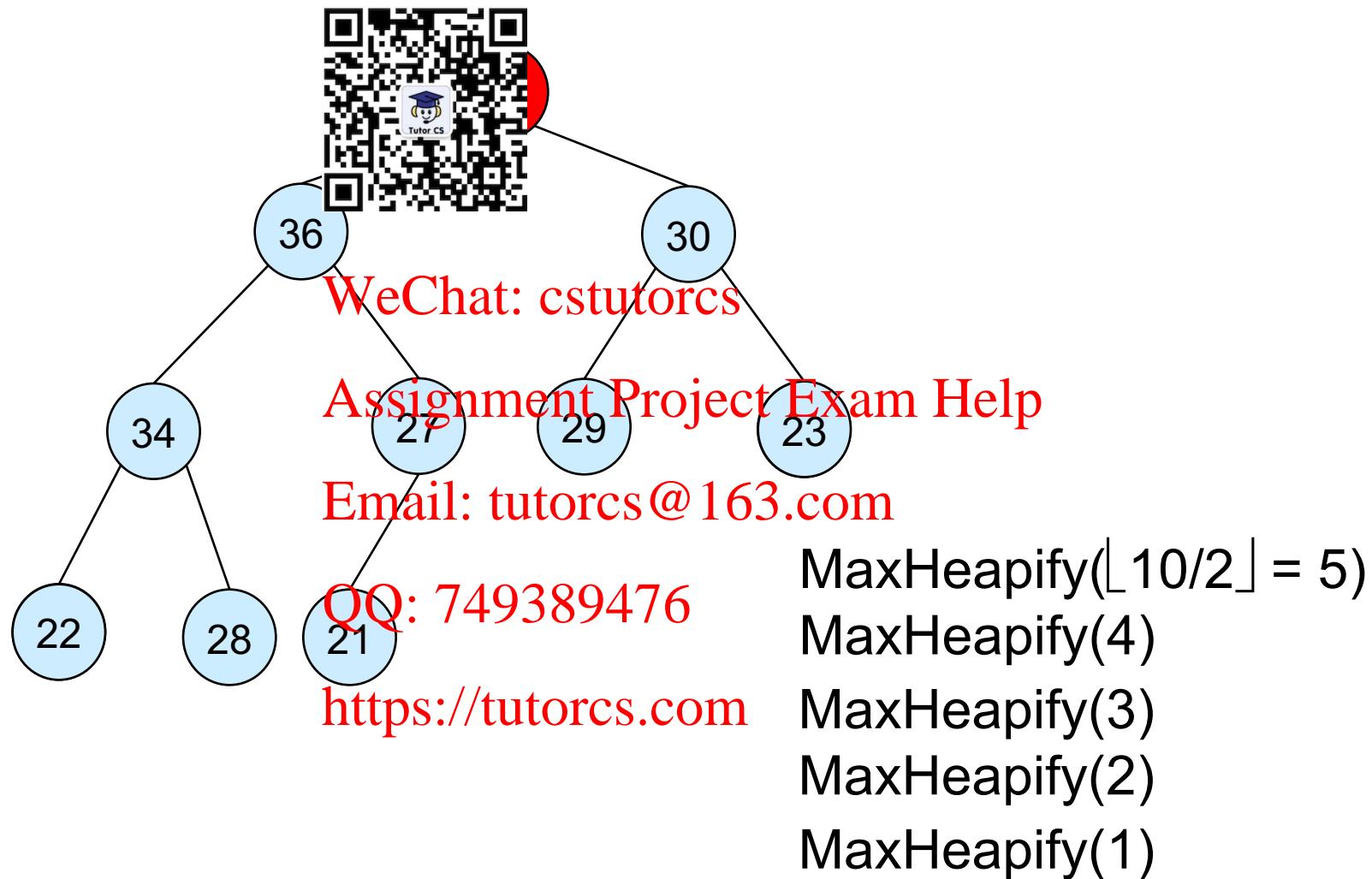
Building a heap - Example

程序代写代做 CS 编程辅导



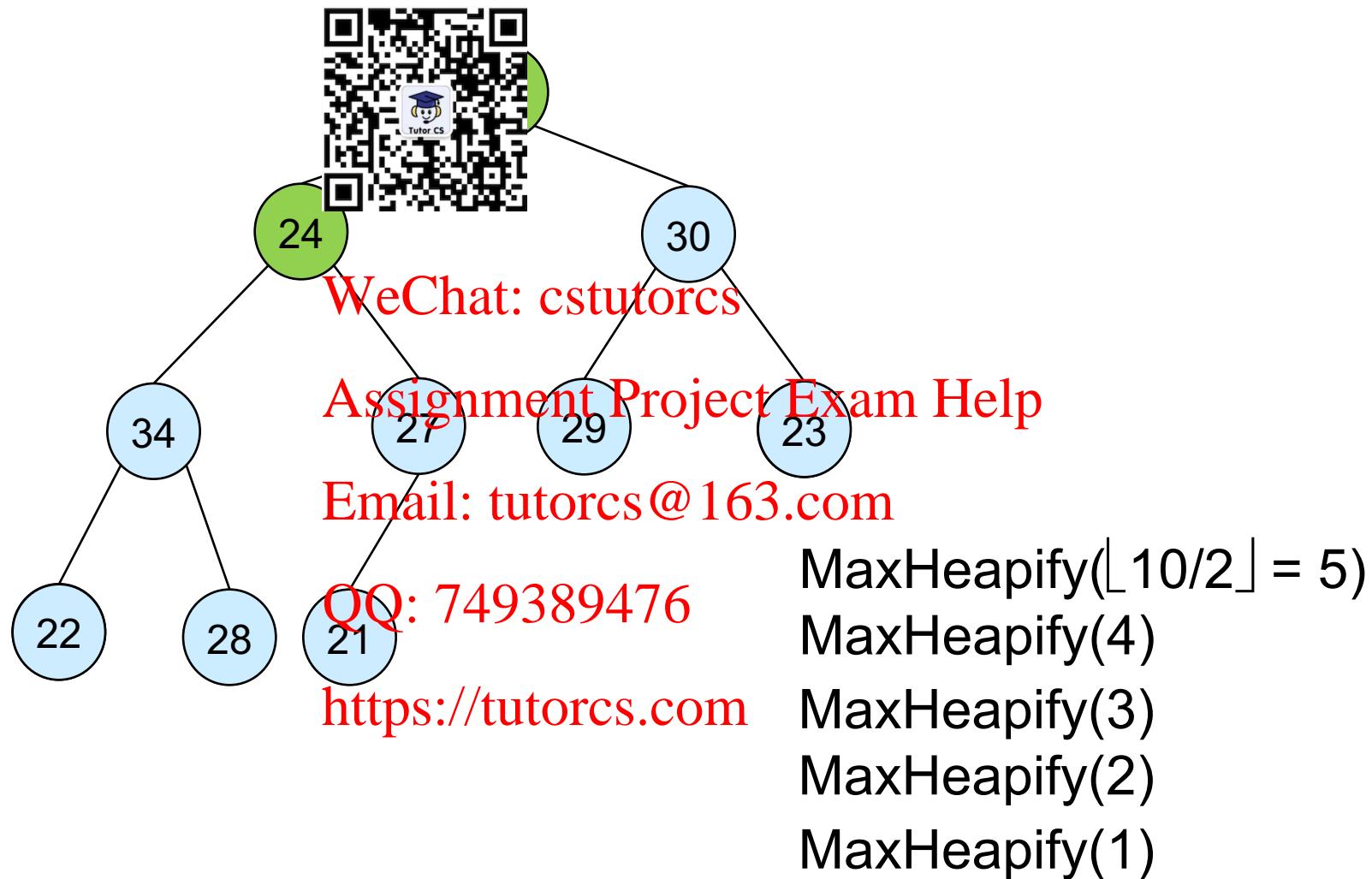
Building a heap - Example

程序代写代做 CS 编程辅导



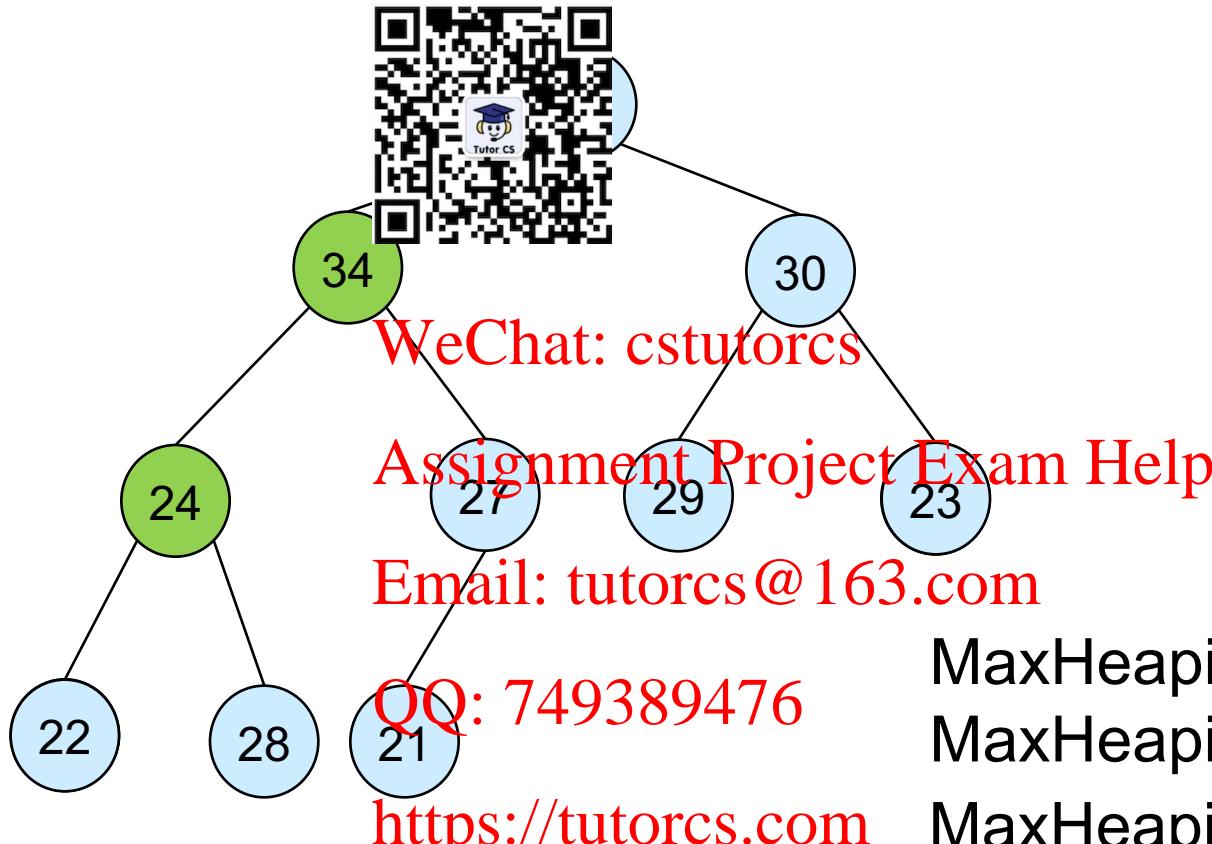
Building a heap - Example

程序代写代做 CS 编程辅导



Building a heap - Example

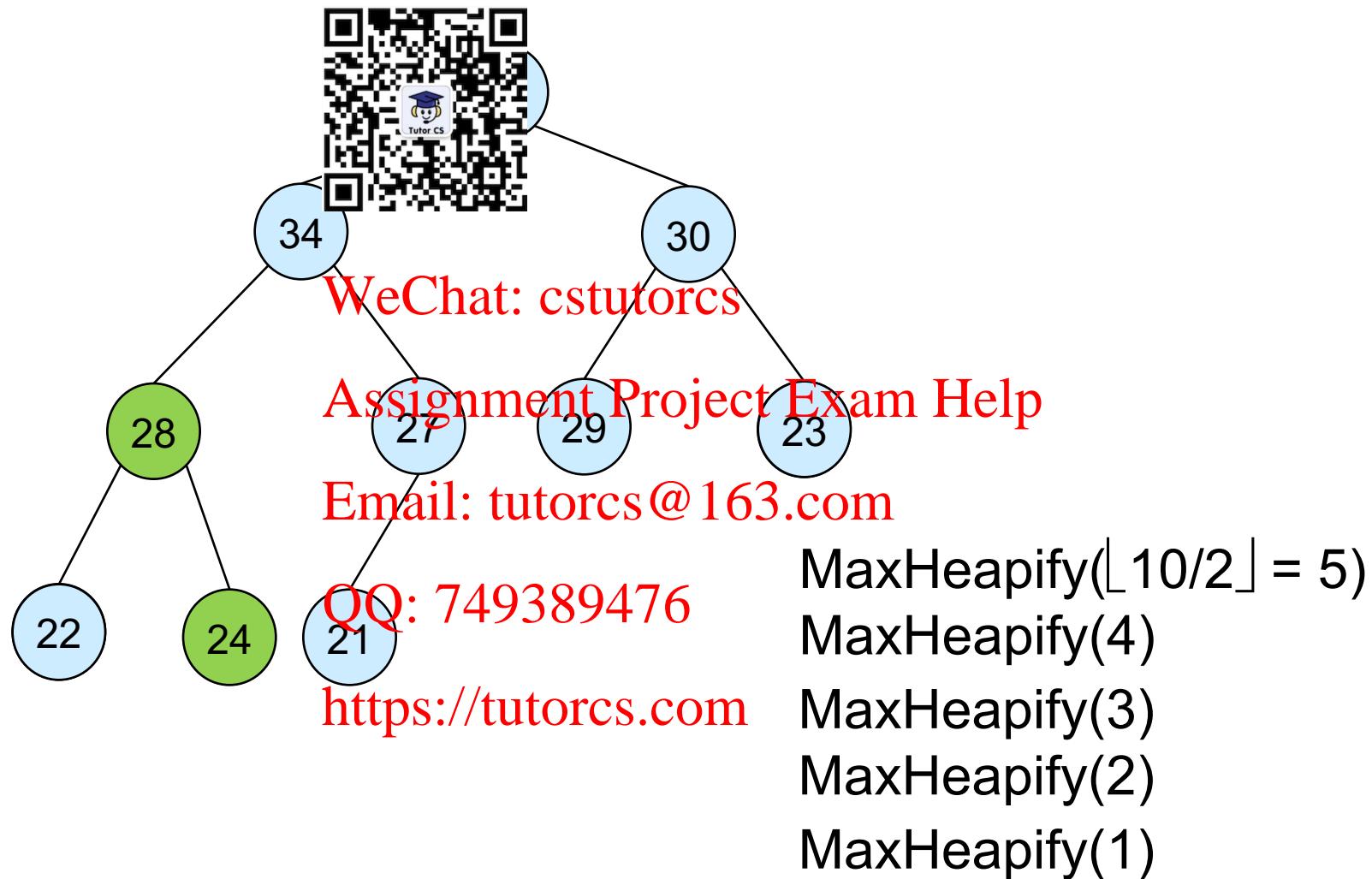
程序代写代做 CS 编程辅导



MaxHeapify($\lfloor 10/2 \rfloor = 5$)
MaxHeapify(4)
MaxHeapify(3)
MaxHeapify(2)
MaxHeapify(1)

Building a heap - Example

程序代写代做 CS 编程辅导



Building a heap - Correctness

程序代写代做 CS 编程辅导

- **Loop Invariant property (LI):** At the start of each iteration of the **for** loop, each node $1, 2, \dots, n$ is the root of a max-heap.
- **Initialization:**
 - Before first iteration
 - Nodes $\lfloor n/2 \rfloor + 1, \lfloor n/2 \rfloor + 2, \dots, n$ are leaves, thus max-heaps.
- **Maintenance:** WeChat: cstutorcs
 - By LI, subtrees at children of node i are max heaps.
 - Hence, MaxHeapify(i) renders node i a max heap root (while preserving the max heap root property of higher-numbered nodes).
 - Decrementing i reestablishes the loop invariant for the next iteration.
- **Termination:** <https://tutorcs.com>
 - bounded number of calls to MaxHeapify
 - At termination, $i = 0$. Then, each node $1, 2, \dots, n$ is the root of a max-heap. In particular, node 1 is.



Assignment Project Exam Help
Email: tutorcs@163.com

QQ: 749389476

Building a heap – Running Time

程序代写代做 CS 编程辅导

- **Loose upper bound:**

- Cost of a *MaxHeapify* is $O(n)$ × # calls to *MaxHeapify*
- $O(\lg n) \times O(n) = O(n \lg n)$



- **Tighter bound:**

WeChat: cstutorcs

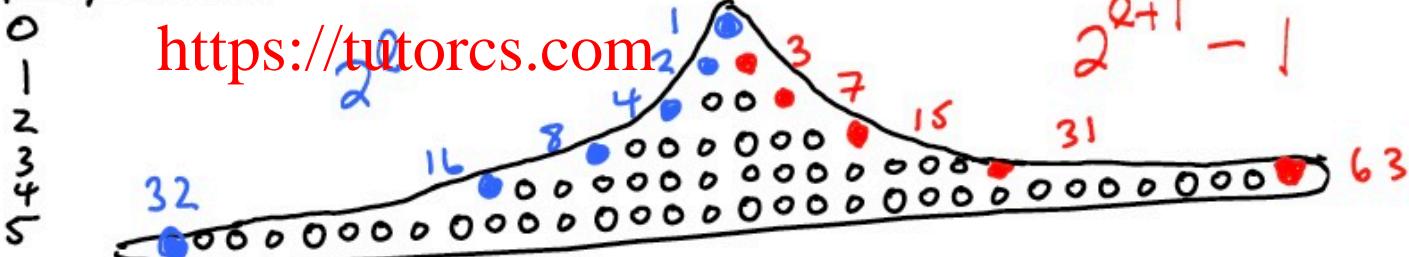
- Cost of *MaxHeapify* is $O(h)$.
- # of nodes with height $h \leq \lceil h/2^{h+1} \rceil$
- Height of heap is $\lfloor \lg n \rfloor$

Taken from Langer2014

e.g. tree height $h = 5$

QQ: 749389476

depth / level



<https://tutorcs.com>

Building a heap – Running Time

程序代写代做 CS 编程辅导

- Tighter bound:

- Cost of MaxHea is $O(h)$.
- #nodes with height h is $\lceil n/2^{h+1} \rceil$
- Height of heap is $\lfloor \lg n \rfloor$

WeChat: cstutorcs



$$\sum_{h=0}^{\infty} \frac{h}{2^h} = \frac{1/2}{(1 - 1/2)^2} = 2$$

Assignment Project Exam Help

$$\sum_{h=0}^{\lfloor \lg n \rfloor} \left\lceil \frac{n}{2^{h+1}} \right\rceil O(h) = O\left(n \sum_{h=0}^{\lfloor \lg n \rfloor} \frac{h}{2^h}\right) = O(n)$$

Email: tutorcs@163.com
QQ: 7493890476

<https://tutorcs.com>

Running time of BuildMaxHeap is $O(n)$

Outline

程序代写代做 CS编程辅导

- Introduction.
- Operations.
- Application.



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Application - Heapsort

程序代写代做 CS 编程辅导

- Combines the better attributes of merge sort and insertion sort.
 - Like merge sort, works well for large inputs. Running time is $O(n \lg n)$.
 - Like insertion sort, simple to implement.
- Introduces an algorithm design technique
 - Create data structure (*heap*) to manage information during the execution of an algorithm.
- The *heap* has other applications beside sorting.
 - Priority Queues (recall COMP250)

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Application - Heapsort

程序代写代做 CS 编程辅导

1. Builds a max-heap from the array.
2. Put the maximum element (the root) at the correct place in the array by swapping it with the element at the last position in the array.
3. “Discard” this last node (note that it is in its correct place) by decreasing the heap size, and call MAX-HEAPIFY on the new root.
4. Repeat this process (goto 2) until only one node remains.



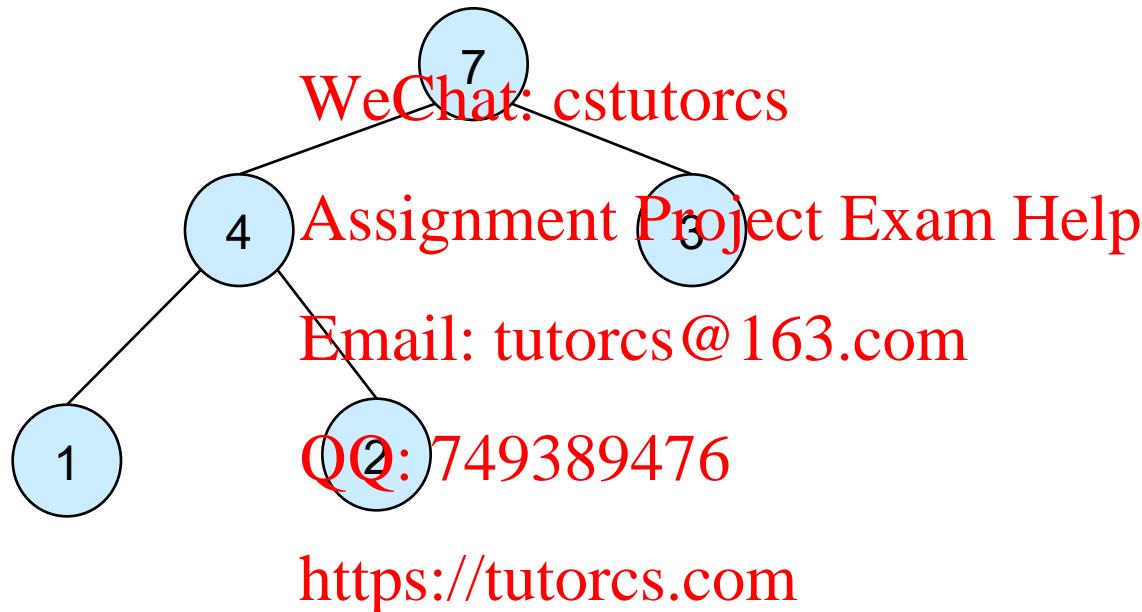
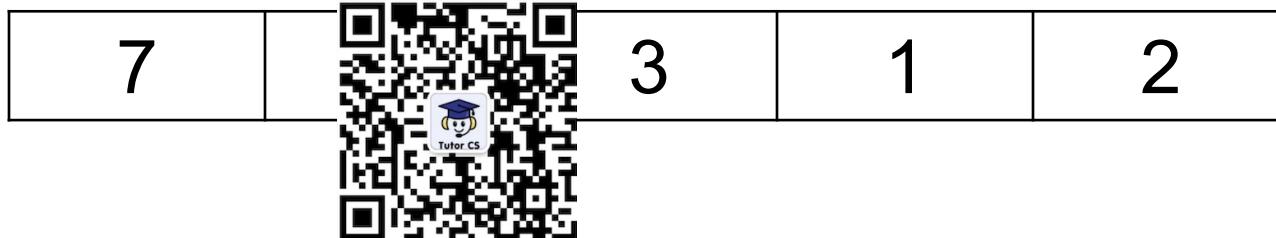
WeChat: cstutorcs

HeapSort(A) Assignment Project Exam Help

1. Build-Max-Heap([Email: tutorcs@163.com](mailto:tutorcs@163.com))
2. **for** $i \leftarrow \text{length}[A]$ **downto** 2
[QQ: 749389476](#)
3. **do** exchange $A[1] \leftrightarrow A[i]$
<https://tutorcs.com>
4. MaxHeapify($A, 1, i-1$)

Heapsort - Example

程序代写代做 CS 编程辅导



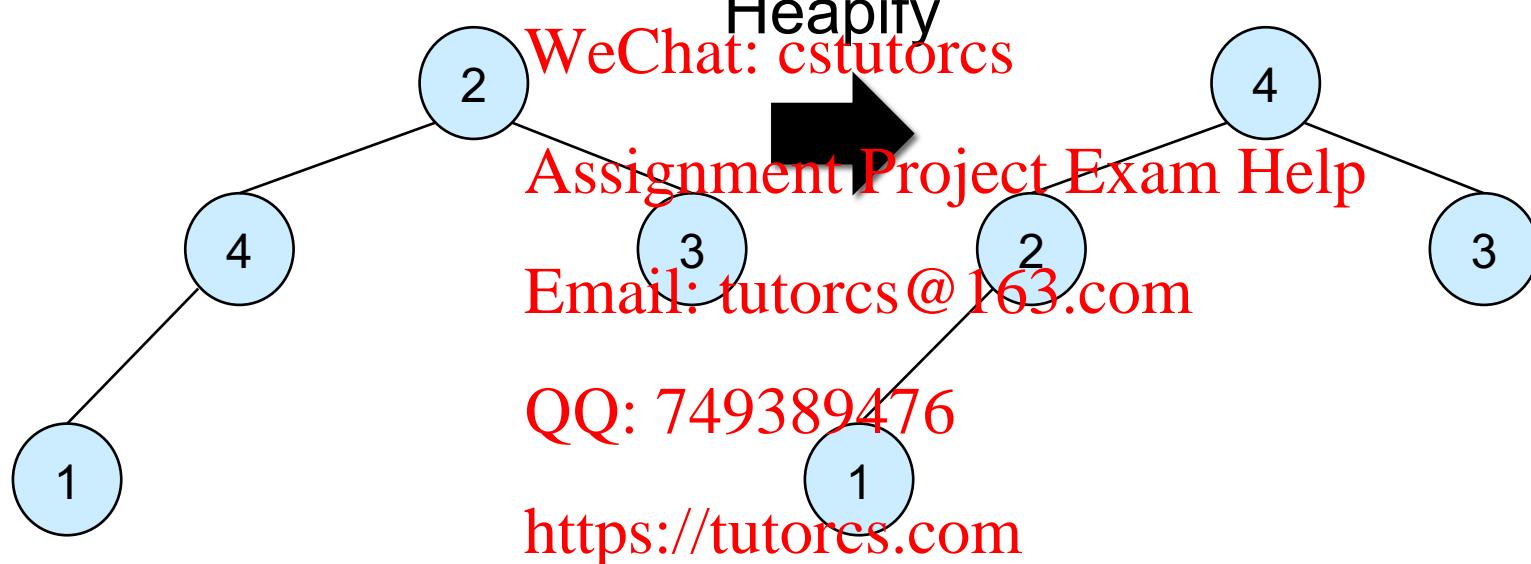
Heapsort - Example

程序代写代做 CS 编程辅导

2	4	3	1
---	---	---	---



4	2	3	1	7
---	---	---	---	---



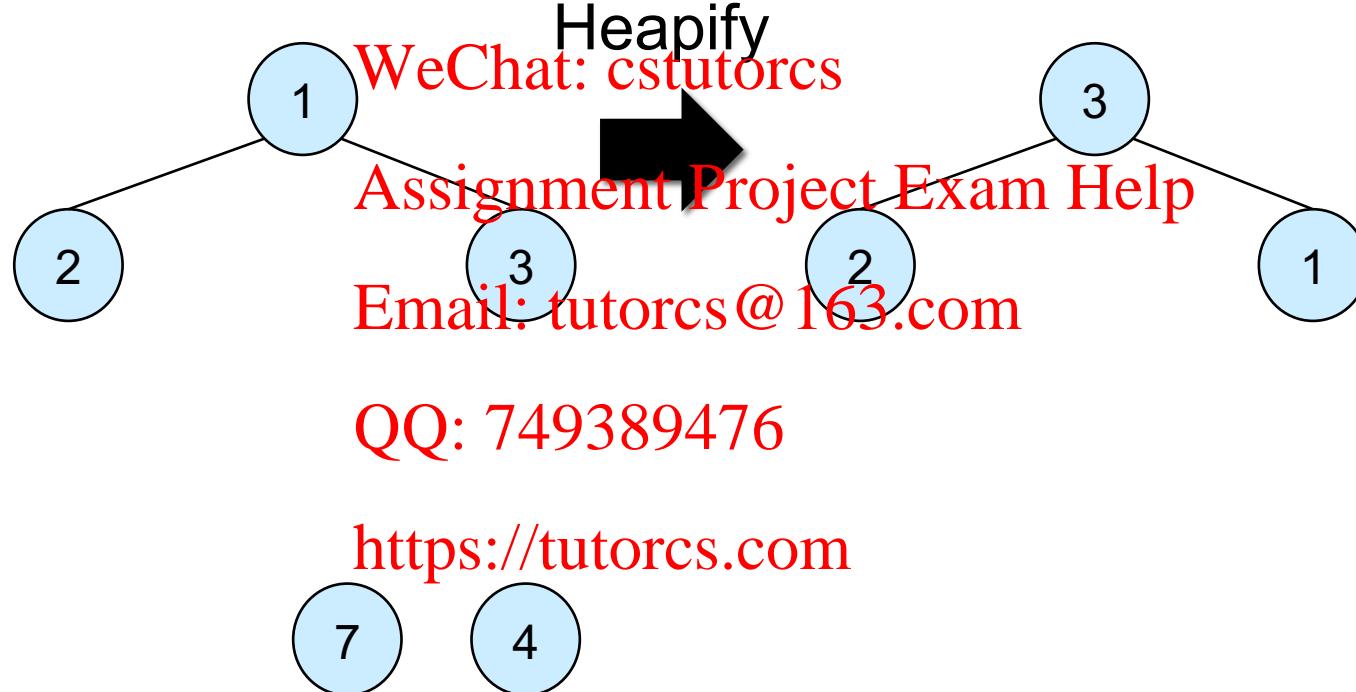
Heapsort - Example

程序代写代做 CS 编程辅导

1	2	3	4
---	---	---	---



3	2	1	4	7
---	---	---	---	---



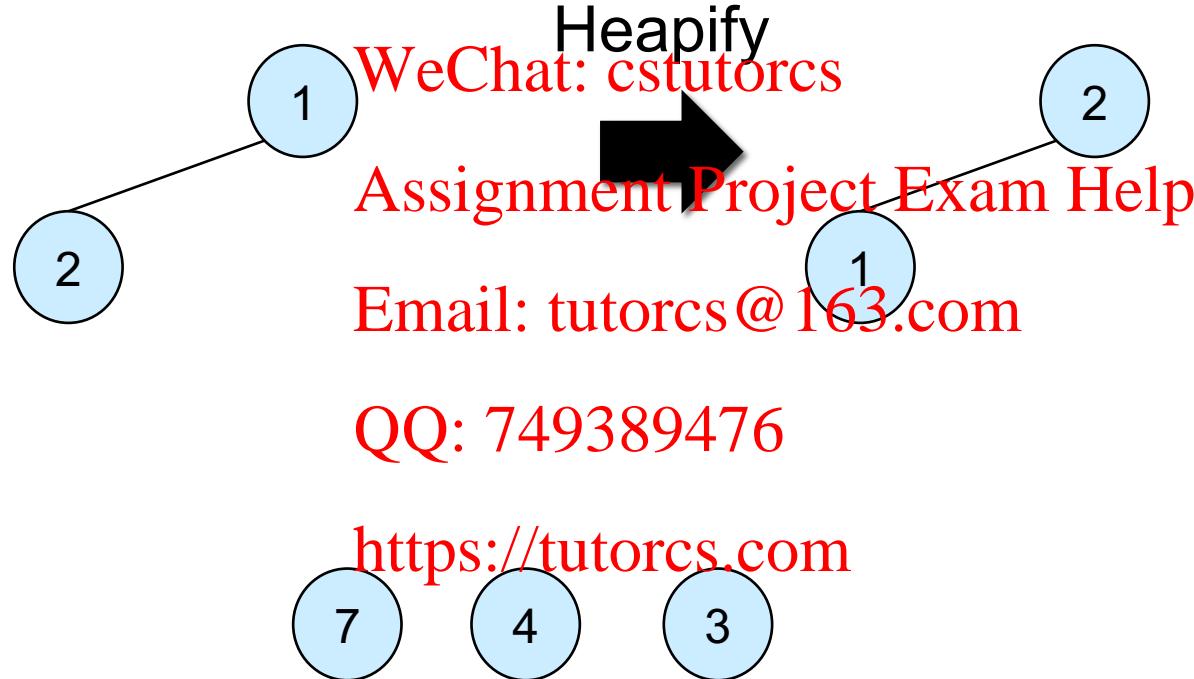
Heapsort - Example

程序代写代做 CS 编程辅导

1	2	3	4
---	---	---	---



2	1	3	4	7
---	---	---	---	---



Heapsort - Example

程序代写代做 CS 编程辅导

1	2	3	4
---	---	---	---



1	2	3	4	7
---	---	---	---	---

1

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

7

4

3

2

Heapsort - Example

程序代写代做 CS编程辅导

- BuildMaxHeap
- for loop $n-1$ times
 - exchange elements $O(1)$
 - MaxHeapify $O(n \lg n)$



=> HeapSort

Assignment Project Exam Help

$O(n \lg n)$

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Outline

程序代写代做 CS编程辅导

- Introduction.
- Operations.
- Application.



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



<https://tutorcs.com>