

Analysis of Algorithms

V. Adamchik

CSCI 570

Lecture 3

University of Southern California

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Reading: chapter 3

Amortized Analysis

In a sequence of operations the worst case does not necessarily occur in each operation ...

Therefore, a traditional worst-case per operation analysis can give overly pessimistic bound.

<https://powcoder.com>

When same operation takes different times, how can we accurately calculate the runtime complexity?

The Aggregate Method

The aggregate method computes the upper bound $T(n)$ on the total cost of n operations.

Assignment Project Exam Help

The amortized cost of an operation is given by $\frac{T(n)}{n}$

<https://powcoder.com>

In this method each operation will get the same amortized cost, even if there are several types of operations in the sequence.

Add WeChat powcoder

$$T(n) = O(1)$$

$$T(n) = O(n)$$

Review Questions ?

2. (T/F) Amortized analysis is used to determine the average runtime complexity of an algorithm.
3. (T/F) Compared to the worst-case analysis, amortized analysis provides a more accurate upper bound on the performance of an algorithm.
4. (T/F) The total amortized cost of a sequence of n operations gives a lower bound on the total actual cost of the sequence.
5. (T/F) Amortized constant time for a dynamic array is still guaranteed if we increase the array size by 5%.
6. (T/F) If an operation takes $O(1)$ expected time, then it takes $O(1)$ amortized time.
7. Suppose you have a data structure such that a sequence of n operations has an amortized cost of $O(n \log n)$. What could be the highest actual time of a single operation?

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

$$n \cdot 1.6 = 60\%$$

$$n+2$$

seq.: $\underbrace{1, 1, 1, \dots, 1}_{n-1}, n^2$. Total cost $= (n-1) + n^2 = O(n^2)$

answer: $O(n \log n)$

$$AC(\text{per increment}) = O(h)/h = O(1)$$

Review: Exercise 2

2. We are incrementing a binary counter, where flipping the i -th bit costs $i + 1$. Flipping the lowest-order bit costs $0 + 1 = 1$, the next bit costs $1 + 1 = 2$, the next bit costs $2 + 1 = 3$, and so on. What is the amortized cost per operation for a sequence of n increments starting from zero?

Assignment Project Exam Help

$\log h - 1$...	1	0
$\log h$		1+1	0+1
2	...	$\frac{n}{2}$	n
MSB			LSB

<https://powcoder.com>

Add WeChat powcoder

bits

cost

flips

$$\text{Cost} = n \cdot 1 + \frac{n}{2} \cdot 2 + \frac{n}{4} \cdot 3 + \dots$$

$$\begin{aligned}
 &+ 2 \cdot \log h = \\
 &= n \sum_{k=0}^{\log h - 1} (k+1) \frac{1}{2^k} \leq \\
 &\leq h \left(\sum_{k=0}^{\infty} \frac{k+1}{2^k} \right) = O(n) \rightarrow \text{const}
 \end{aligned}$$

The Accounting Method

The accounting method (or the banker's method) computes the individual cost of each operation.

Assignment Project Exam Help

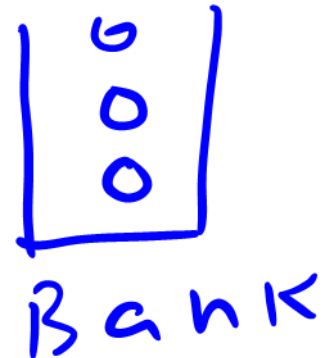
We assign different charges to each operation; some operations may charge more or less than they actually cost.

<https://powcoder.com>

Add WeChat powcoder

The amount we charge an operation is called its amortized cost.

$O(1)$ token



~~$O(1) = ?$~~ **Discussion Problem**

surplus
Hill
Bank

You have a stack data type, and you need to implement a FIFO queue. The stack has the usual POP and PUSH operations, and the cost of each operation is 1. The FIFO has two operations: ENQUEUE and DEQUEUE.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

We can implement a FIFO queue using two stacks. What is the amortized cost of ENQUEUE and DEQUEUE operations?

Queue engine A. push

dequeue

$O(3)$ ~~$O(1)$~~ AC

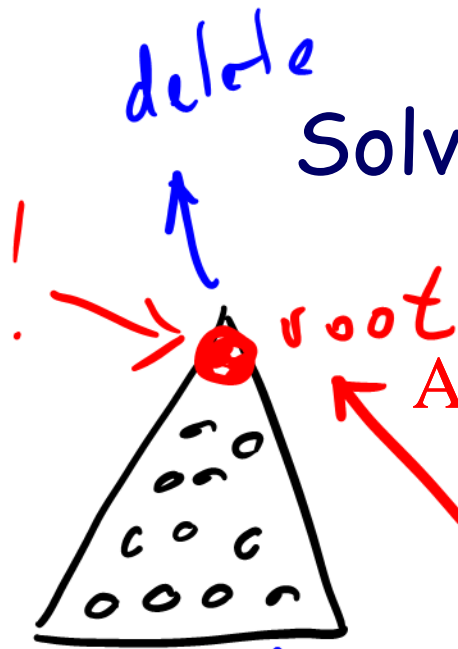
$O(1)$ ~~$O(3)$~~ AC

$O(1)$ B. pop

Bank 3, 0!

Heap and Priority Queue for

Solving Optimization Problems



root

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

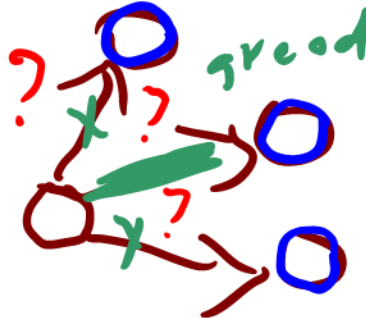
1. sort $O(n \log n)$

2. find min $O(n)$

3. heap $O(\log n)$

Heap - implementation
PQ - interface, API

insert



the best choice

insert
delete

BST
 $L < P < R$

max $\rightarrow P \geq L \ \& \ P \geq R$
 Binary min-Heap

A binary heap is a complete binary tree which satisfies the heap ordering property.

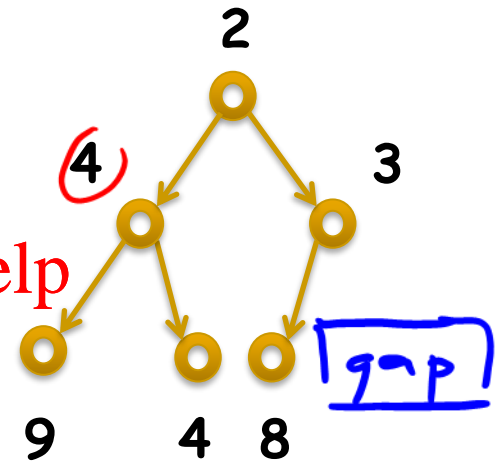
Assignment Project Exam Help

1. Structure Property

<https://powcoder.com>

2. Ordering Property

$P \leq L \ \& \ P \leq R$



0	1	2	3	4	5	6	7
X	2	4	3	9	4	8	9 < P

Consider k -th element of the array,

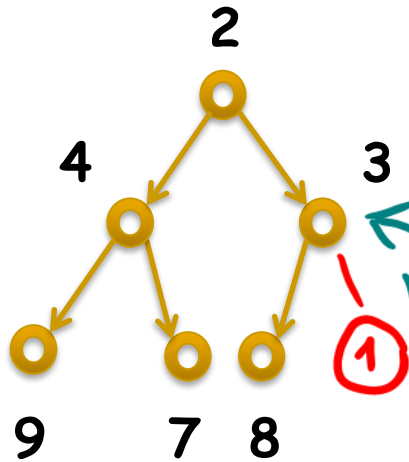
- its left child is located at $2*k$ index
- its right child is located at $2*k+1$ index
- its parent is located at $k/2$ index

$\rightarrow 5/2 = 2$

①

insert (tree reps)

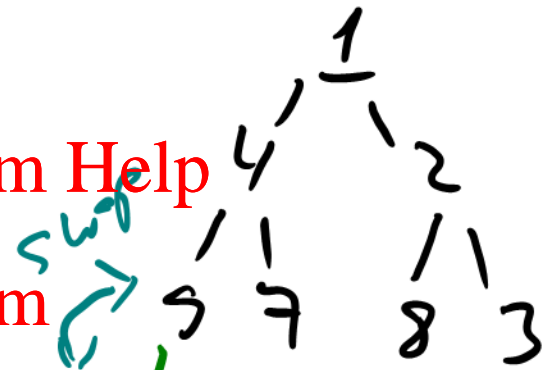
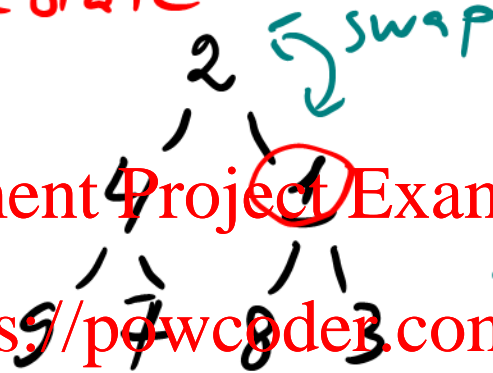
- a) into a gap
- b) percolate up (bubble up)



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



insert 5

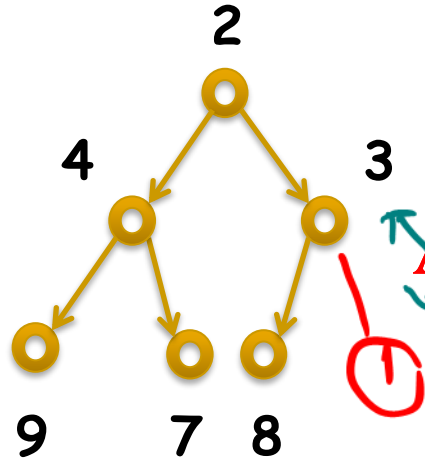
Runtime (worst-case) =

= # of swaps = height of the heap

= $O(\log n)$, here n is the input size

implementation insert (array reps)

① - append



index

index	0	1	2	3	4	5	6	7
	X	2	4	3	9	7	8	1
	X	2	4	1	9	7	8	3
	X	1	4	2	9	7	8	3

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

single for-loop

Discussion Problem 1

Prove it by construction.

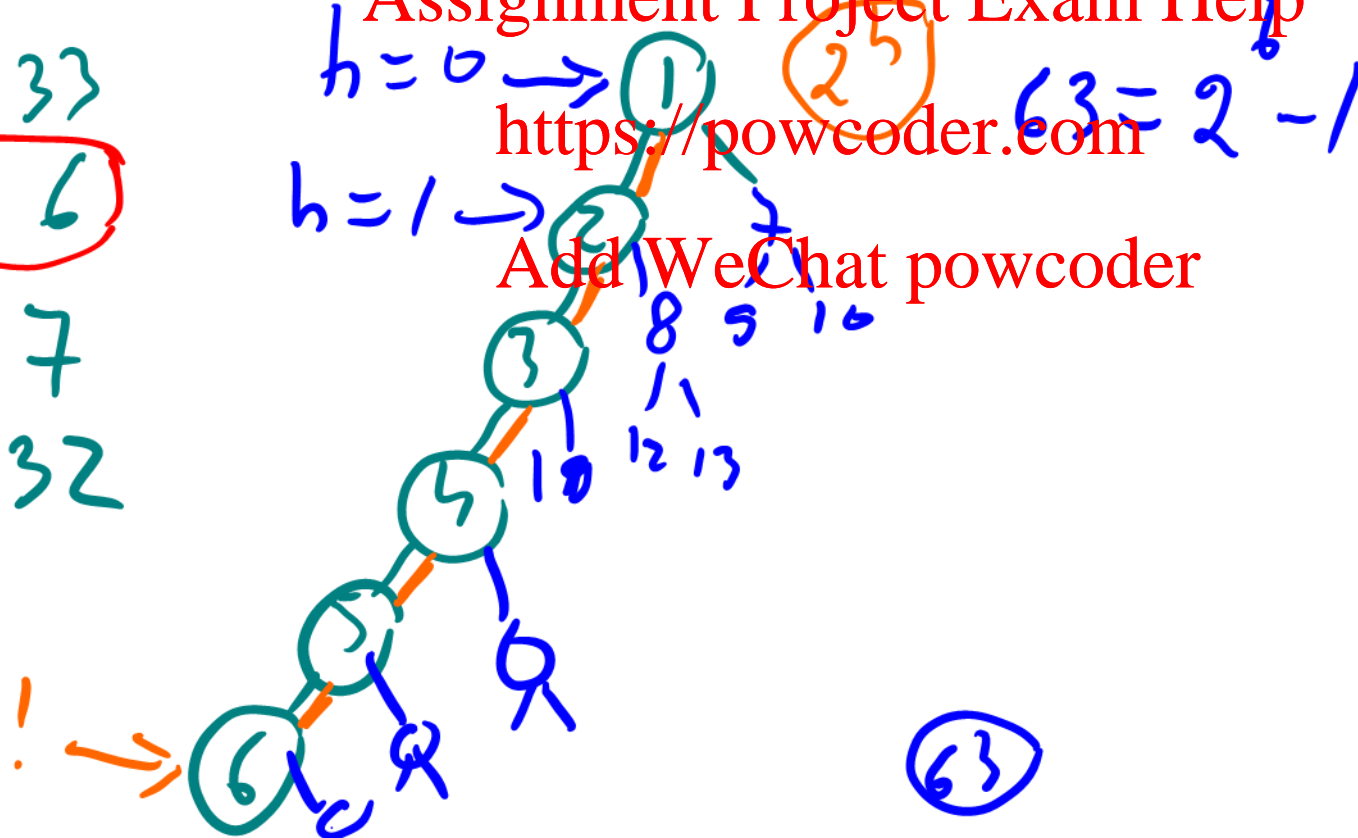
The values 1, 2, 3, ..., 63 are all inserted (in any order) into an initially empty min-heap. What is the smallest number that could be a leaf node?

Assignment Project Exam Help

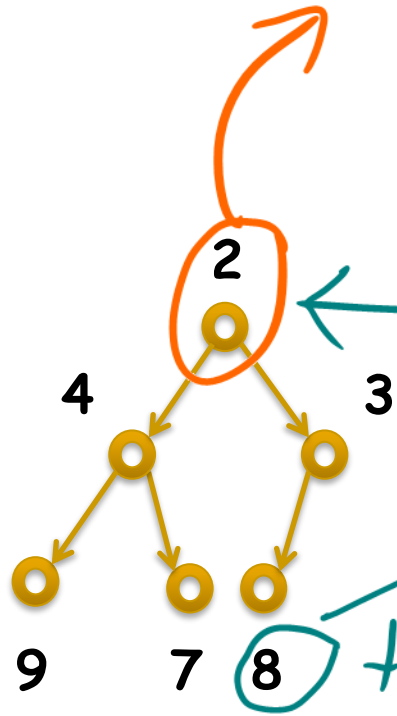
<https://powcoder.com>

Add WeChat powcoder

- a) 33
- b) 6**
- c) 7
- d) 32



delete the root
deleteMin (tree reps)

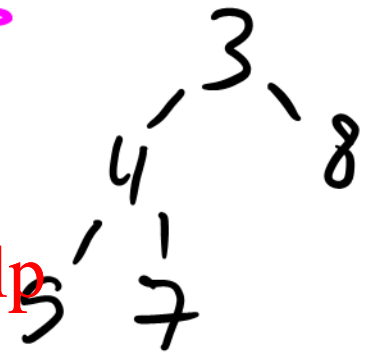
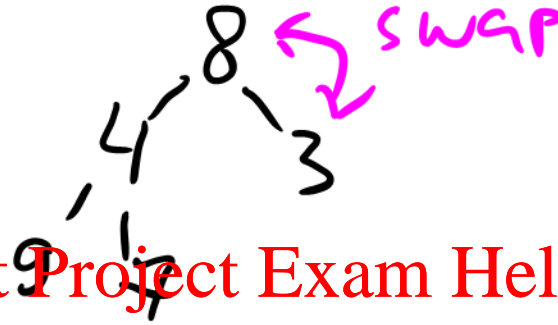


Assignment Project Exam Help

<https://powcoder.com>

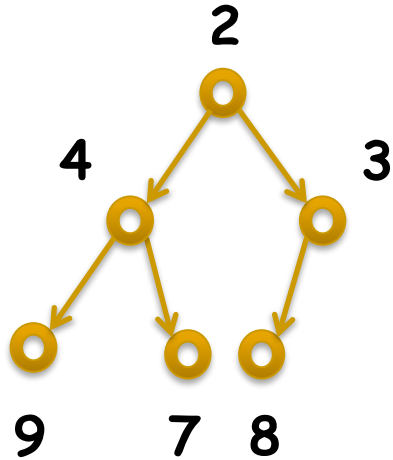
Add WeChat powcoder

Worst-case = $\# \text{ of swaps} = O(\log n)$



percolate down

implementation deleteMin (array reps)



0	1	2	3	4	5	6	7
X	2	4	3	9	7	8	
X	8	4	3	9	7		
X	3	4	8	9	7		

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

for-loop

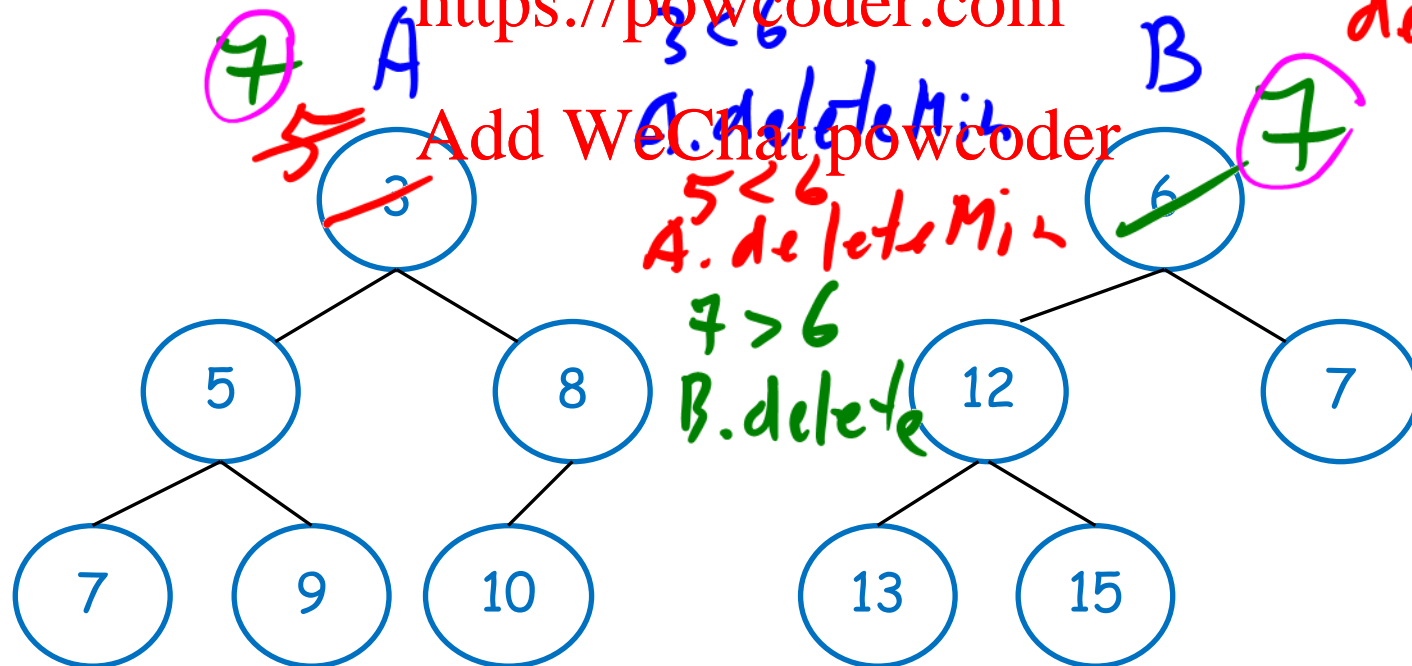
$G \xrightarrow{BFS} \text{spanning tree}$ Discussion Problem 2

Suppose you have two binary min-heaps, A and B, with a total of n elements between them. You want to discover if A and B have a key in common. Give a solution to this problem that takes time $O(n)$. Do not use the fact that heaps are implemented as arrays.

Assignment Project Exam Help

<https://powcoder.com>

API: insert
deleteMin



Build a Heap by Insertion

Given an array - turn it into a heap.

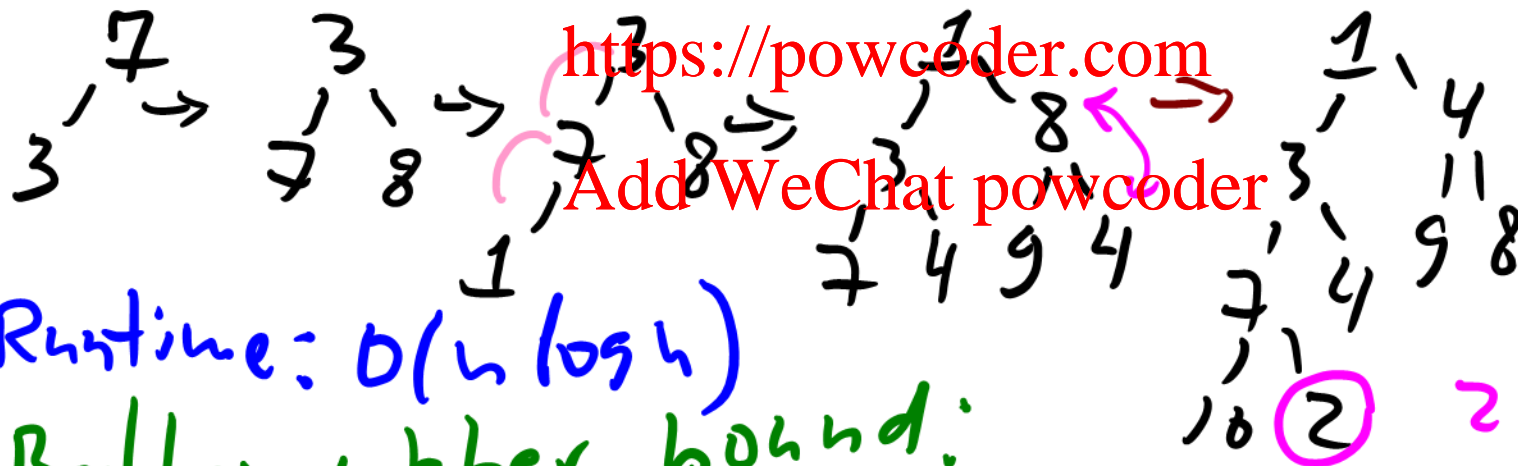
input-size n

input: 7, 3, 8, 1, 4, 9, 4, 10, 2, 0

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Runtime: $O(n \log n)$

Better upper bound:

$$\log 2 + \log 3 + \log 4 + \dots + \log n = \log(n!) =$$

(see lect. 1) $= \Omega(n \log n)$

2 swaps

Compute the # of swaps!

Build a Heap in $O(n)$

Heapify:

7, 3, 8, 1, 4, 9, 4, 10, 2, 0

top-bottom

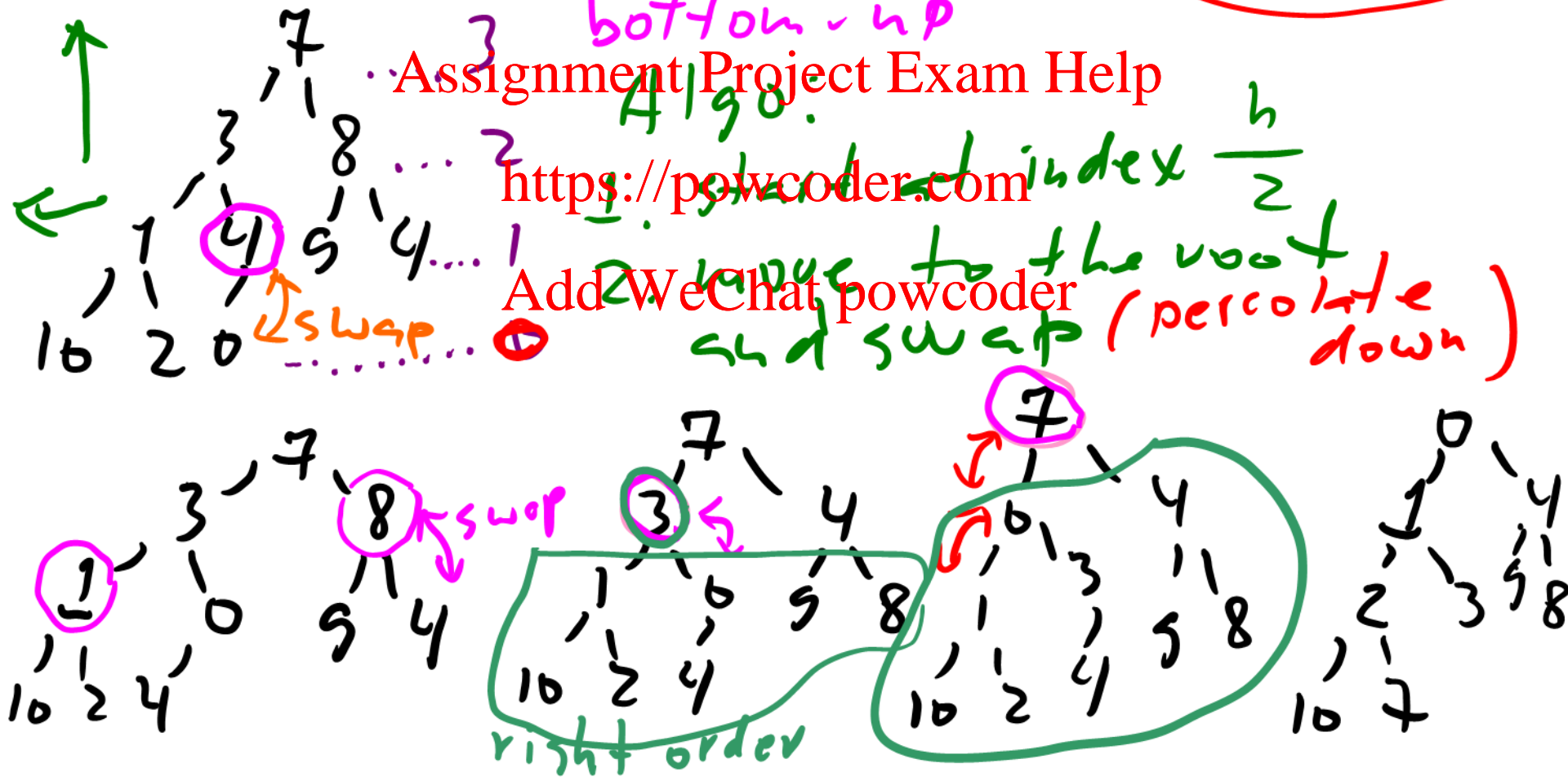
bottom-up

Assignment Project Exam Help

<https://powcoder.com>

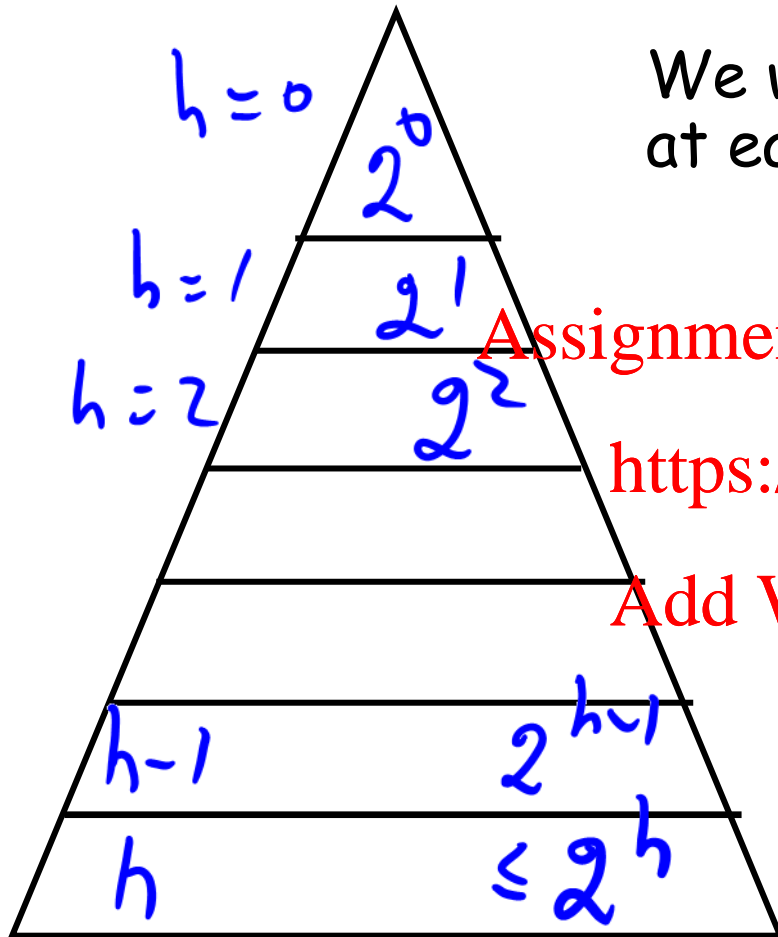
Add WeChat powcoder

move to the root and swap (percolate down)



Complexity of heapify $h = \log n$

We will count the max number of swaps at each level



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

height	# of nodes	# of swaps
0	1	$\leftrightarrow h$
1	2	$\leftrightarrow h-1$
2	4	$\leftrightarrow h-2$
...
$h-1$	2^{h-1}	$\leftrightarrow 1$

\sum

Total # swaps (# of comparisons)
 Complexity of heapify

$$\underbrace{1 \cdot h}_{k=h} + \underbrace{2 \cdot (h-1)}_{k=h-1} + \underbrace{4 \cdot (h-2)}_{k=h-2} + \dots + \underbrace{2^{h-1} \cdot 1}_{k=1} =$$

$$= \sum_{k=1}^h 2^{h-k}$$

Assignment Project Exam Help

<https://powcoder.com>

$$\leq 2^h \sum_{k=1}^{\infty} \frac{1}{2^k} = 2^h \cdot O(1) = 2^{\log h} = O(h)$$

→ const

$$h = \log n$$

linear
time

Building a BST (by insertion): $n \cdot n = O(n^2)$

Discussion Problem 3

heapsort

How would you sort using a binary heap?

What is its runtime complexity?

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Algo: 1. run heapify once, $O(n)$
2. in a loop: delete Min
Runtime: $O(n + n \cdot \log n) = O(n \log n)$

Sort with a BST (given an array)
Algo: 1. Build a BST, $O(n^2)$

2. in-order traversal, $O(n)$

$O(n^2)$

2, 5, 2, 7
4, B

STABLE: 2_A, 2_B, 5, 7

HEAPSORT

no extra space

in-place

nonstable ??

Run delMin n-times

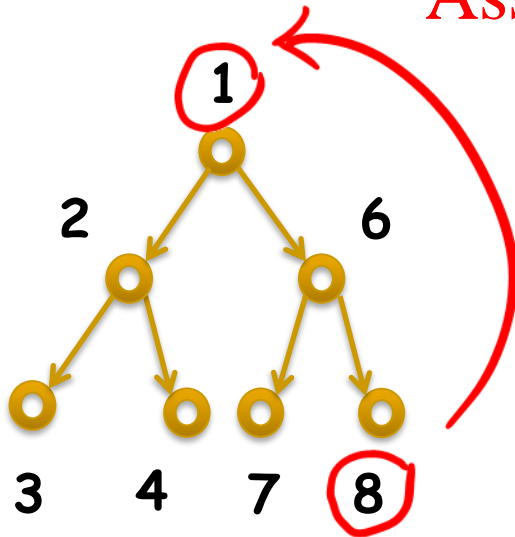
$O(n \log n)$



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



0	1	2	3	4	5	6	7
	1	2	6	3	4	7	8

	2	3	6	8	4	7	1
--	---	---	---	---	---	---	---

	3	4	6	8	7	2	1
--	---	---	---	---	---	---	---

	4	8	6	7	3	2	1
--	---	---	---	---	---	---	---

sorted part

Discussion Problem 4

A, B

How would you merge two binary min-heaps?

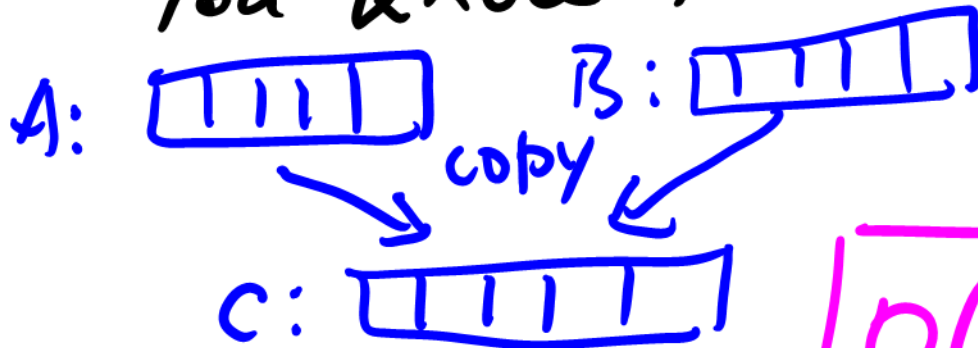
What is its runtime complexity?

Assignment Project Exam Help

1. API: $B.insert(A.deleteMin)$, $O(n \log n)$
<https://powcoder.com>

Add WeChat powcoder

2. Library implementation.
You know that a heap is an array



a) copy A, B to C
b) run heapify on C

$O(n)$

$K = n/2$

$K = 5$

Discussion Problem 5

Devise a heap-based algorithm that finds the k -th largest element out of n elements. Assume that $n > k$. What is its runtime complexity?

A. Offline Algorithm (all data available)

1. build max-heap of size n , $O(n)$
2. delete Max k times, $O(k \cdot \log n)$

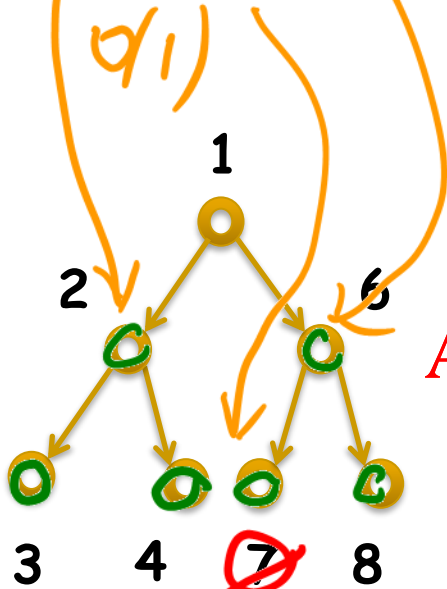
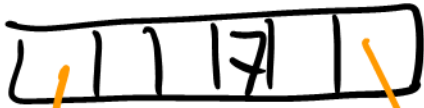
Add WeChat powcoder

B. Online Algorithm (streaming data)

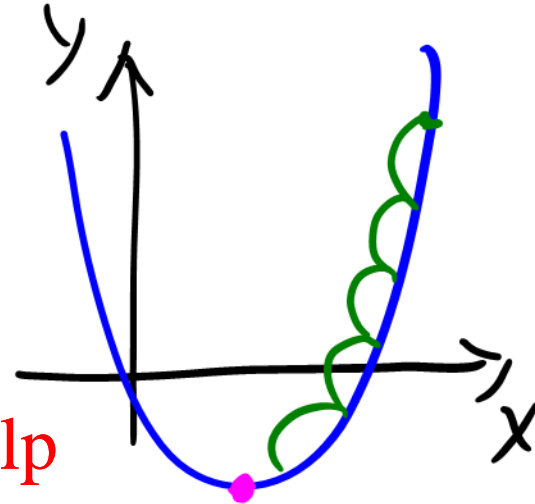
1. build min-heap of size K , $O(K)$
2. wait for $(K+1)$
3. root $< (K+1)$ item, delete Min, insert
- root $> (K+1)$ item, nothing

$O(K + (n-K) \log K)$

hash table



$O(\log n)$
decreaseKey
for min-heap



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

percolate up, $O(\log n)$

must be

$O(1)$???

Algo: 1. find (7) ? ~~$O(n)$~~
2. replace by 2 and percolate up
 $O(\log n)$