CS4311 Design and Analysis of Algorithms

Tutorial: Assignment 2

speaker: 劉向瑄(Jenny)

outline

- Part 1: explaining assignment 2 and some hints
- Part 2: solution of assignment 1

Question 1

- $1 + 2 + \ldots + n = \theta(n^2)$
- there is a trap in the proof!

Question 1 Hint

- check definition of O-notation

Question 2

- insert operation is correct:

  after insert operation, both the shape property and heap property should be satisfied

Question 2 Hint

- We offer a **wrong** proof here:

prove by induction:

Basis: insert l to a heap which height is 0, after insertion the shape and heap properties are both satisfied

Question 2 Hint

- We offer a **wrong** proof here:

  Inductive step: insert l to heap whose height is k, it will be two cases,
  (1) the heap height grow up to k+1
  (2) the heap height is still k
  In both cases, l will rise until it's parent is smaller than it, that is, nodes in the subtree rooted on l are all bigger than l, heap property is satisfied. At the beginning of insert operation, we put inserted data to the proper position(shape property is satisfied!) So the insert operation is correct.

Question 3

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
3	4	1	2	6	5	10	8	7	9

1	2	3	4	5	6	7	8	9	10

the kth smallest number is stored at A[j]
with k-d <= j <= k+d
d is a distance parameter < n

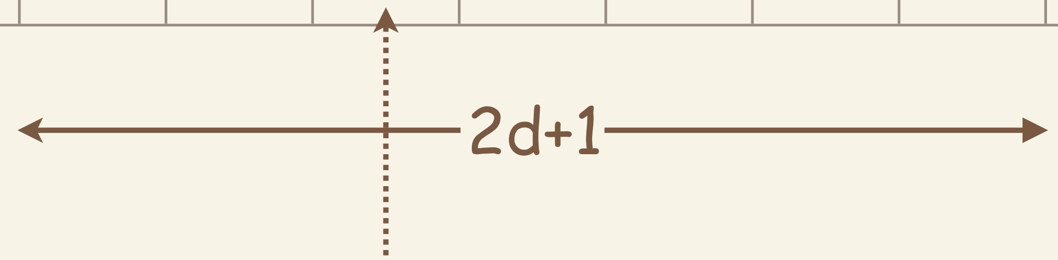
Question 3

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
3	4	1	2	6	5	10	8	7	9

6th smallest number

Question 3

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
3	4	1	2	6	5	10	8	7	9



6th smallest number

Question 3

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
3	4	1	2	6	5	10	8	7	9

1st smallest number

Question 3

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
3	4	1	2	6	5	10	8	7	9

d+1

1st smallest number

Question 3 Hints

- Your task is sorting the nearly sorted sequence and get completely sorted one in $O(n \log d)$-time

Question 3 Hints

- Hint 1:

sort n data needs O(n logn) time
sort 2n data needs O(n logn) time, too

merge sort needs O(n logn) time whether we merge two subsequences or three for each step
($O(n \log_2 n) = O(n \log n)$
$O(n \log_3 n) = O(n \log n)$)

Question 3 Hints

- Hint 2: association of log n
  - problem size halved
  - comparison sort
  - extract-min in a heap
  - binary search

Hint of Hints!

- Our hints are not the only way to solve the questions!
- Solving problems by your own ideas is always highly encouraged!

(but it must be correct, of course)