

Grader Output

ProblemSet2_S

45.00 / 45.00

points earned

8 / 8 autograded

cells passed

Graded Cells

Cell -1 (cell- c21e18266dbaa2k

Passed | 0 / 0

points

[View feedback](#)

Cell -1 (cell- 691fd6725fc83928

Passed | 0 / 0

points

[View feedback](#)

Cell 2 (cell- 6956a1e5f9f6578e

Passed | 10.00 /

10.00 points

[View feedback](#)

Cell 4 (cell- 2d85778743b6729

Passed | 15.00 /

15.00 points

[View feedback](#)

Problem Set # 2 (Basic Datastructures and Heaps)

Topics covered:

- Basic data-structures
- Heap data-structures
- Using heaps and arrays to realize interesting functionality.

Problem 1 (Least-k Elements Datastructure)

We saw how min-heaps can efficiently allow us to query the least element in a heap (array). We would like to modify minheaps in this exercise to design a data structure to maintain the **least k** elements for a given $k \geq 1$ with $k = 1$

being the minheap data-structure.

Our design is to hold two arrays:

- (a) a sorted array A of k elements that forms our least k elements; and
- (b) a minheap H with the remaining $n - k$ elements.

Our data structure will itself be a pair of arrays (A, H) with the following property:

- H must be a minheap

Cell 4 (cell-3366a32bb62b3cc

Passed | 0 / 0

points

[View feedback](#)

Cell 4 (cell-99c28c7d7bb49cc

Passed | 0 / 0

points

[View feedback](#)

Cell 6 (cell-0f15a9f966d07a21

Passed | 5.00 /

5.00 points

[View feedback](#)

Cell 8 (cell-dbf8d2bf71617a5c

Passed | 15.00 /

15.00 points

[View feedback](#)

- A must be sorted of size k .
- Every element of A must be smaller than every element of H .

The key operations to implement in this assignment include:

- insert a new element into the data-structure
- delete an existing element from the data-structure.

We will first ask you to design the data structure and then implement it.

(A) Design Insertion Algorithm

Suppose we wish to insert a new element with key j into this data structure. Describe the pseudocode. Your pseudocode must deal with two cases: when the inserted element j would be one of the least k elements i.e, it belongs to the array A ; or when the inserted element belongs to the heap H . How would you distinguish between the two cases?

- You can assume that heap operations such as $\text{insert}(H, \text{key})$ and $\text{delete}(H, \text{index})$ are defined.
- Assume that the heap is indexed as $H[1], \dots, H[n - k]$ with $H[0]$ being unused.
- Assume $n > k$, i.e, there are already more than k elements in the data structure