CSE310 HW03, Thursday, 02/25/2021, Due: Friday, 03/05/2021

Please read the instructions carefully. **You have to use the companion answer sheet (which is a fillable PDF file) to type/select your answers to the questions described here**. Hand-written assignment (or photo of it) will not be graded. **Submit the filled PDF file of the answer sheet on Gradescope, following the link on Canvas**. You should name your file using the format **CSE310-HW03-LastName-FirstName.pdf**. **Make sure that your submission can be viewed clearly on gradescope for auto-grading.** Adobe Acrobat Reader can be found at `https://get.adobe.com/reader/`.

Q1 (15 points) A max-heap with `capacity` 20 and `size` 10 is shown in the following array format. The following three sub-questions all refer to this max-heap (not the heap you obtained after doing some operations).

i	1	2	3	4	5	6	7	8	9	10
$A[i]$	29	27	25	23	21	19	17	15	13	11

(a) On the answer sheet, show the result after applying `heap-extract-max(A)` to the max-heap at the start of this question.

(b) On the answer sheet, show the result after applying `heap-increase-key(A, 10, 28)` to the max-heap at the start of this question.

(c) On the answer sheet, show the result after applying `max-heap-insert(A, 29)` to the max-heap at the start of this question.

Q2 (15 points) In class, we have studied max-heap and its operations in details. The min-heap data structure is defined similarly, `with max replaced by min, greater than replaced by less than, etc`. The operations of min-heap is also symmetric to the corresponding operations of the max-heap. This question is about the min-heap. A min-heap with `capacity` 20 and `size` 10 is shown in the following array format. The following three sub-questions all refer to this min-heap (not the heap you obtained after doing some operations).

i	1	2	3	4	5	6	7	8	9	10
$A[i]$	11	13	15	17	19	21	23	25	27	29

(a) On the answer sheet, show the result after applying `heap-extract-min(A)` to the min-heap at the start of this question.

(b) On the answer sheet, show the result after applying `heap-decrease-key(A, 9, 10)` to the min-heap at the start of this question.

(c) On the answer sheet, show the result after applying `min-heap-insert(A, 9)` to the min-heap at the start of this question.

Q3 (15 points) This question is related to disjoint set operations. Assume that we are using union by rank and find with path compression. Suppose that you are given a disjoint set structure described by the following array. The following three sub-questions all refer to **this disjoint set** (not the disjoint set you obtained after doing some operations).

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$A[i]$	-3	1	1	3	1	5	5	7	-3	9	9	11	9	13	13	15

(a) On the answer sheet, show the result after applying `union(16, 4)` to the disjoint set at the start of this question.

(b) On the answer sheet, show the result after applying `union(6, 14)` to the disjoint set at the start of this question.

(c) On the answer sheet, show the result after applying `find-set(7)`, `find-set(15)` to the disjoint set at the start of this question.

Q4 (8 points) On the answer sheet, answer the following questions.

(a) What is the worst-case time complexity of `build-heap` on n elements?

(b) What is the worst-case time complexity of `extract-max` on a max-heap with n elements?

(c) What is the worst-case time complexity of `insertion` onto a max-heap with n elements?

(d) What is the worst-case time complexity of `increase-key` onto a max-heap with n elements?

(e) What is the worst-case time complexity of `find-set` in a disjoint set data structure, assuming we are using union by rank and find with path compression?

(f) What is the worst-case time complexity of `link` in a disjoint set data structure, assuming we are using union by rank and find with path compression?

(g) What is the worst-case time complexity of `union` in a disjoint set data structure, assuming we are using union by rank and find with path compression?

(h) What is the worst-case time complexity of m operations of `union`, `find`, and `make-set`, including n `make-set` operations at the start?