Illinois Institute of Technology
Department of Computer Science

Honesty Pledge

CS 535 Design and Analysis of Algorithms
Fall Semester, 2015

Fill out the information below, sign this sheet, and submit it with Homework 0.

I promise, *on penalty of failure of CS 535*, not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others on the homework problems in CS 535.

All work I submit will be mine and mine alone.

I understand that *all* resources in print or on the web, aside from the text and class notes, used in solving the homework problems *must be explicitly cited.*

PRASANTH BHAGAVATULA	BPrasanth	A20355611	8/27/2015
Name (printed)	Signature	Student ID	Date

ASSIGNMENT: HW0 CS535 - Fall 2015

NAME: PRASANTH BHAGAVATULA

CWID: A20355611

1.(a) Does the corrupted code still work (that is, correctly find the i^{th} smallest element) always, some times, or never? Explain

Ans: The corrupted code will work some times. ~~As long~~ Consider, a scenario where in the RANDOMIZED - PARTITION (A, p, r) will always return 'r'. In this case, 'q' will be same as 'r' and 'k' will be '$r-p+1$', which is nothing but the length of array. Assuming, we are not finding maximum element of the array, $i < k$, and so it will call RANDOMIZE - SELECT (A, p, r, i). It will go in ~~a loop~~ an infinite loop.

ASSIGNMENT: HW0 CS535 – Fall 2015

NAME: PRASANTH BHAGAVATULA

CWID: A20355611

1.(b) Analyze the worst-case running time of the corrupted code.

Ans. As considering the scenario, explained in the 1.(a); in the worst case scenario the algorithm can go in an infinite loop and the running time cannot be determined.

ASSIGNMENT: HW0 CS535 - Fall 2015

NAME: PRASANTH BHAGAVATULA

CWID: A20355611

1.(c) Analyze the best case running time of the corrupted code

Ans: The best case would be when the value of 'K' is same as 'i' and so the statement number 6 will get executed and the corrupted code will not run, even once. In this case, the running time of this algorithm mainly depends on the running time of RANDOMIZED-PARTITION (A, p, r) and so the best case running time of this algorithm would be $O(n)$.

ASSIGNMENT: HW0 CS535 - Fall 2015

NAME: PRASANTH BHAGAVATULA

CWID: A20355611

1.(d) Analyze the average-case running time of the corrupted code.

Ans: Consider the expected values of actual RANDOMIZED-SELECT algorithm.

$$E[T(n)] \leq \sum_{k=1}^{n} \frac{1}{n} \cdot E[T(\max(k-1, n-k))] + O(n)$$

Here, we ~~have~~ are calculating 'max(k-1, n-k)', because we ~~alreak~~ already found the ~~correct~~ k^{th} order statistic and since it is different from 'i', we are going for recursive call of the same algorithm. Because of the corrupted code, the expected value would be

$$E[T(n)] \leq \sum_{k=1}^{n} \frac{1}{n} \cdot E[T(\max(k, n-k))] + O(n)$$

Consider the expression $\max(k, n-k)$.

$$\max(k, n-k) = \begin{cases} k & \text{if } k \geq \lceil n/2 \rceil \\ n-k & \text{if } k < \lceil n/2 \rceil \end{cases}$$

(cont.)

So, here, even in this ~~sm~~ summation we will find the same kind of terms that we see for the actual ~~(a)~~ correct RANDOMIZED-SELECT i.e. if n is even, $T(\lceil n/2 \rceil)$ upto $T(n-1)$ appears twice and if n is odd $T(\lfloor n/2 \rfloor)$ also appears in the ~~sm~~ summation. So the relation is same as of the ~~ai~~ correct RANDOMIZED-SELECT and so the average-case running time of this corrupted code will also be the same as the correct code, which is $O(n)$

ASSIGNMENT: HW0 CS 535 - Fall 2015
NAME: PRASANTH BHAGAVATULA
CWID: A20355611

1.(c) There is something strange about your answer to the previous part; what is strange and how do you explain it?

Ans. The strange thing is that the average case running time for both the corrupted code and actual code of RANDOMIZED-SELECT is same. This would be because, we are just passing one extra-element in the array (RANDOMIZED-SELECT(A, p, q, i)); ie A[q] which is greater than any of elements in A[p ... q-1] and so finding the i^{th} smallest element in A[p ... q-1] and A[p .... q] does not have any difference in asymptotic running time.