

Preliminary Quiz

Please answer all the questions.

- Given a set S of n distinct integers,
 1. describe an algorithm to compute the second largest integer in S .
 - A ready solution would be to do two linear scans to find it, taking $O(n)$ time. You cannot do better asymptotically, but if you are actually bothered with minimizing the exact number of comparisons, you can do so by holding a tournament, and then finding the winner among the $\log_2 n$ losers who lost to the actual winner.
 2. describe an algorithm to compute the median of S . For simplicity, assume n is an odd integer. Then, note that the median of S is the number in S that is smaller than exactly half the numbers in S .
 - For those who are not too thick into algorithm already, a $O(n \log n)$ solution (by sorting the n numbers and then choosing the middle element) would be quite acceptable. In fact, if you are not familiar with sorting, you can even give an $O(n^2)$ algorithm of doing $\frac{n+1}{2}$ linear scans on the input element, every time computing the minimum and setting it aside after the scan is over. The minimum of the $\frac{n+1}{2}$ -th linear scan would be the median. However, you can actually do better. You can either use the **Median of Medians** algorithm by Blum or use an intelligent partitioning scheme as used in quicksort. In both cases it is actually possible to get the median in $O(n)$ time. The proofs are quite complicated and would rather be presented in detail later when we would cover the **Quicksort** and **Selection** algorithms in class.
- In probability theory, the **birthday paradox** states that in a group of 23 (or more) randomly chosen people, there is more than 50% probability that some pair of them will have the same birthday. For 57 or more people, the probability is more than 99%, although it cannot be exactly 100% unless there are at least 366 people.

In order to test this theory, we design an algorithm to check whether there exists two students in a class that have the same birthday. We suggest two algorithms:

- For every pair of students in the class, I check whether the two students have the same birthday.
 - I pass a calendar around, I ask every student to mark his/her birthday on the calendar and let me know if his/her birthday is already marked by another student.
1. Discuss which one of these algorithms take less time?
 - Since there are ${}^n C_2$ pairs, which is $O(n^2)$, the first algorithm takes more time as the second algorithm is clearly $O(n)$ -time algorithm.
 2. Discuss which one of these algorithms use less memory space?
 - The first one needs just a constant amount of scratch space to store the two birthdays for comparing them (or, their indices when the elements are stored in an array). On the other hand, the second one takes $O(k)$ scratch space, where k is the universe of possible birthdays (which is 365 in our case). Clearly the first algorithm is more space efficient.

- Consider N the universe of firstnames. For the sake of simplicity, assume that each firstname is either a male name (like John, Patrick etc.) or a female name (like Sara, Sally etc.) – it cannot be an unisex name. Let C be the class of all sets which contain only male firstnames or contain only female firstnames. Let C also include the empty set.

- Is the union of any pair of sets in C also in C ?

No, as $\{\text{John}\} \cup \{\text{Sara}\} = \{\text{John}, \text{Sara}\}$ is clearly not in C .

- Is the intersection of any pair of sets in C also in C ?

Yes.

- Is the complement of any set in C also in C ?

No. For example, the complement of $\{\text{John}\}$ would be a the set containing all the female firstnames as well as all the male firstnames except John. This set would clearly not be in C .

I.e, is C closed under the operations of intersections, unions and complement respectively?

– It is closed only under intersection.

- Suppose the following sentence is true.

“If I attended all the lectures, I will get an A in this course.”

What can I conclude logically from the each of the following 2 cases:

- I got an A

Cannot say anything about the attendance, since there could be other factors which also help getting an A .

- I did not get an A .

It is a fact that I did **not** attend all the lectures, as then I would have surely gotten an A .

- How many distinct subsets are there of the set $\{1, 2, 3, \dots, n\}$? Your expression should be in terms of n .

- The powerset of any set of n distinct elements has 2^n distinct subsets (including both the null set and the super set). This is because every subset can be created by making a decision of either including (or not including) each member of the superset. Since there are n elements and 2 possible decisions for each element, there could be 2^n such distinct outcomes.