CS 161 **Problem Set 3**
Winter 2020 **Due:** Wednesday, January 29, 2020, at 10am on Gradescope

Note: Because of MLK day, we only had one lecture this week... so we have a commensurately short problem set! This problem set will be worth half of a normal problem set in your final grade.

Exercise

Please do the exercises on your own.

1. **(12 pt.)** In this exercise, we'll explore different types of randomized algorithms. We say that a randomized algorithm is a **Las Vegas algorithm** if it is always correct (that is, if it returns, then it returns the right answer), but the running time is a random variable. We say that a randomized algorithm is a **Monte Carlo algorithm** if there is some probability that it is incorrect. For example, QuickSort (with a random pivot) is a Las Vegas algorithm, since it always returns a sorted array, but it might be slow if we get very unlucky.

   The alien population from the previous PSETs is back, and you want to solve the same problem as you did on PSET 2. However, this time, you'd like to use randomized algorithms. As a reminder, there are n aliens on their planet, of many different species, but you're guaranteed that there is one species which makes up a strict majority of the aliens. You can't tell the aliens apart, but there's an expert alienologist who can, and she can answer queries to `isTheSame(p1,p2)` which returns True if `p1` and `p2` are aliens of the same species, and False otherwise. Your goal is to find a member of the majority species. Consider the three algorithms seen in Figure 1, all of which call the subroutine isMajority (also seen in Figure 1), and all of which attempt to find a majority alien.

   Fill in the table below. If it is helpful, the LaTeXcode for the table is copied at the end of HW3.

Algorithm	Monte Carlo or Las Vegas?	Expected running time	Worst-case running time	Probability of returning a majority alien
Algorithm 1				
Algorithm 2				
Algorithm 3				

   [**We are expecting:** *Your filled in table. You must use asymptotic notation for the running times, as usual ignoring constants and lower order terms. For the probability of returning a majority alien, give the* ***tightest*** *bound that you can given the information provided (i.e., your bound should still hold with the* ***worst case*** *inputs that still meet the conditions outlined above). No justification is required.*]

```
Algorithm 1: findMajorityAlien1
Input: A population $P$ of $n$ aliens
while true do
    Choose a random $p \in P$;
    if isMajority(p) then
        return $p$;
```

```
Algorithm 2: findMajorityAlien2
Input: A population $P$ of $n$ aliens
for 100 iterations do
    Choose a random $p \in P$;
    if isMajority(p) then
        return $p$;
return $P[0]$;
```

```
Algorithm 3: findMajorityAlien3
Input: A population $P$ of $n$ aliens
Put the aliens in $P$ in a random order.;
/* Assume it takes time $\Theta(n)$ to put the $n$ aliens in a random order     */
for $p \in P$ do
    if isMajority(p) then
        return $p$;
```

```
Algorithm 4: isMajority
Input: A population $P$ of $n$ aliens and an alien $p \in P$
Output: True if $p$ is a member of a majority species
count $\leftarrow 0$;
for $q \in P$ do
    if isTheSame(p,q) then
        count ++;
if $count > n/2$ then
    return True;
else
    return False;
```

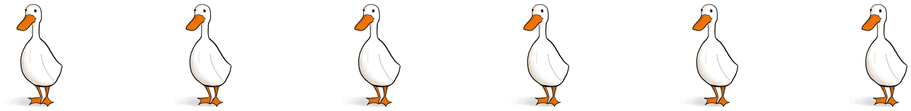
Figure 1: Three randomized algorithms for finding a majority alien

Problem

You may talk with your fellow CS161-ers about the problems. However:

- Try the problems on your own *before* collaborating.
- Write up your answers yourself, in your own words. You should never share your typed-up solutions with your collaborators.
- If you collaborated, list the names of the students you collaborated with at the beginning of each problem.

2. **(10 pt.)** [**Duck ice cream party**] After all that hard work dancing in PSET 2, the troupe of dancing ducks is ready to relax and celebrate. Suppose that the n ducks are standing in a line.



Each duck has a favorite flavor of ice cream: vanilla, chocolate, or vegan sorbet. You'd like to sort the ducks so that all the chocolate-loving ducks are on the left, the vanilla-loving ducks are on the right, and the vegan-sorbet-loving ducks are in the middle.[1] You can only do two types of operations on the ducks:

Operation	Result
`ask(j)`	Ask the duck in position j about its favorite ice cream flavor
`swap(i,j)`	Swap the duck in position j with the duck in position i

The ducks are eager to get their ice cream, but each of the above operations takes around 5 seconds to execute. Also, you didn't bring a piece of paper or a pencil, so you can't write anything down and have to rely on your memory. Like many humans, you can remember up to seven integers[2] between 0 and $n-1$ at a time.

(a) **(7 pt.)** Design an algorithm to sort the ducks which takes $O(n)$ seconds, and requires you to remember no more than seven integers[3] between 0 and $n-1$ at a time.

[**We are expecting:** *Pseudocode **AND** a short English description of your algorithm.*]

(b) **(3 pt.)** Justify why your algorithm is correct, why it takes only $O(n)$ seconds, and why it requires you to remember no more than seven integers at a time.

[**We are expecting:** *Informal justifications of the correctness, runtime, and memory usage of your algorithm that are both clear and convincing to the grader. If it's easier for you to be clear, you can give a formal proof of correctness, but you do not have to. It is okay to appeal to the correctness of an algorithm that we have seen in class, as long as you clearly explain the relationship between the two algorithms.*]

[1] Since the ducks will be eating the ice cream right out of the carton, it's easier if they are sorted by their preferred ice cream flavor.

[2] see, e.g., `https://en.wikipedia.org/wiki/The_Magical_Number_Seven,_Plus_or_Minus_Two`

[3] You don't need to use all seven storage spots, but you can if you want to. Can you do it with only two?

Feedback

There's no "correct" answer here, and it is completely anonymous.

3. **(1 pt.)** Please fill out the following poll, which asks about sections so far:

   `https://forms.gle/SczmDGTsMmCC45qw6`

   Did you fill out the poll?

   [**We are expecting:** *The answer "yes."*]

Helpful LaTeXcode!

Here is the code that generated the table for the alien exercise:

```
\begin{center}
\begin{tabular}{|c|p{3cm}|p{2cm}|p{2cm}|p{4cm}|}
\hline
Algorithm & Monte Carlo or Las Vegas? & Expected running time
& Worst-case running time & Probability of returning a majority alien \\
\hline
\textbf{Algorithm 1} &  &  &  & \\
\hline
\textbf{Algorithm 2} &  &  &  & \\
\hline
\textbf{Algorithm 3} &  &  &  & \\
\hline
\end{tabular}
\end{center}
```