# CPSC 320 2020S2: Tutorial 6

## 1 Who's the boss?

Recently, Hooli Inc, a large international company, had an election to choose their CEO for the next 5-year period. A candidate can be selected as the new CEO, only if they gather a majority vote, meaning that from the $n$ votes submitted by shareholders, they get more than $n/2$ votes. Hooli claimed that in the recent vote no candidate got a majority, so the current CEO will be hired for the time being. A freedom of information request was made by a news agency to Hooli to disclose the votes for the election.

Hooli said that, due to privacy concerns, they cannot disclose the votes because the reputation of the candidates with very few votes will be damaged. Instead, they have been able to quickly set up an *API* for the news agency. Every query to the API is of the form $QueryAPI(i, j)$, where $i$ and $j$ are two integers in the range $[1..n]$ (the number of votes, $n$ is known publicly). The query returns True if the $i$th vote and the $j$th vote were cast for the same candidate, and False otherwise.

As an example of how to access the API, we can run $QueryAPI(1, k)$ $n - 1$ times with $k = 2, \ldots, n$ to count how many of the votes are given to the same candidate as the first vote. Hooli is charging a ridiculous amount of money for every query made to the API. We are interested in checking whether there is a majority in the $n$ votes or not, and we want to use the least number of queries.

1. Suppose that we partition the list $S$ of $n$ votes, $S$, into two lists, say $S_l$ and $S_r$, of votes with approximately $n/2$ votes each. The following claim is not correct - how can you change the statement to be correct?

   **Claim**: Suppose that candidate $X$ has a majority in list $S$. Then candidate $X$ has a majority in *both* $S_r$ and $S_l$.

Claim:

1. If candidate X has a majority in S, then candidate X has a majority in both $S_r$ and $S_l$.

→ Few variants are correct

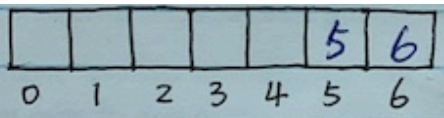
- Suppose that candidate X has a majority in list S. Then at least one of $S_r$ and $S_l$ have a majority of X.
- Suppose that candidate X has a majority in list S. Then the majority of $S_r$ is X OR the majority of $S_l$ is X.
- Suppose that candidate X has a majority in list S. Then ($S_r$ has a majority and majority of $S_r$ is X.

## 2 Divide & conquer

Consider the problem of taking a **sorted** array $A$ containing **distinct** (and not necessarily positive) integers, and determining whether or not there is a position $i$ such that $A[i] = i$.

1. Describe a divide-and-conquer algorithm to solve this problem. Your algorithm should return such a position if it exists, or *nil* otherwise. If $A[i] = i$ for several different integers $i$, then you may return any one of them.

2. Analyze the running time of your algorithm as a function of the number of elements of $A$.

2. A 

Binary search

① If A[mid] ≤ mid, then $j \geq 0$ ($j \in \mathbb{Z}$). $j \leq mid$, $A[mid-j] < mid-j$.

Proof: Base case: when $j=0$, Then it's true, $A[mid] < mid$

IS: Assume it holds for $k \in [0, j]$, we ~~prove~~ can prove that $j+1$ still holds.

$A[mid-(j+1)] \leq A[mid-j]-1 < (mid-j)-1 = mid-(j+1)$.

So $A[mid-(j+1)] < mid-(j+1)$.

② If $A[mid] > mid$, then $A[mid+j] > mid+j$.

```
func FindPos(A, first, last)
    if first = last:
        if A[first] = first
            return first
        return null
    mid = ⌊(first + last)/2⌋
    if A[mid] = mid
        return mid
    if A[mid] < mid
        return FindPos(A, mid+1, last)
    else
```