Homework 5

RT Hatfield

19 September 2016

1.
- An individual merge has complexity $O(l+m)$, where l and m are the sizes of the lists being merged. We have k lists of n elements to merge, so the first merge has complexity $O(2n)$, the second has complexity $O(2n+n)$, the third has complexity $O(3n+n)$, and so forth. We end up with a complexity like $O(\sum_{i=1}^{k-1}(i+1)n)$, which reduces to the category of $O(kn)$.
- Divide and conquer? I'll divide you!

  Really, though, I've been thinking about this and not sure how to go about it. I keep coming up with solutions that involve taking pairs of the lists and merging them, but they all end up seeming to run in $O(kn)$ time.

2.
- If the majority elements of A_1 and A_2 are the same, obviously that's also the majority element of A. If not, whichever majority element is larger is the majority element of A.

  The idea behind what I'm doing here is that if any element takes up over half the list, then that element is going to have to be repeated sequentially at some point. I find any elements which are repeated sequentially and save those, ditching the rest.

```
Data: A list $A$
Result: The majority element, if any, of $A$; null if none
if $A.size == 2$ then
    if $A[0] == A[1]$ then
        return $A[0]$;
    end
    else
        return null;
    end
end
else
    $A_1 = A[0 \dots \frac{n}{2}]$;
    $A_2 = A[(\frac{n}{2} + 1) \dots n]$;
    $a$ = check($A_1$);
    $b$ = check($A_2$);
    if $a == b$ then
        return $a$;
    end
    else
        if $a == null$ then
            return $b$;
        end
        return $a$;
    end
end
```

- This is based on the same insight I had when working out the previous algorithm, only it decides a little differently with less overhead.

```
Data: A list $A$
Result: The majority element, if any, of $A$; null if none
if $A.size == 2$ then
    if $A[0] == A[1]$ then
        return $A[0]$;
    end
    else
        return null;
    end
end
else
    for $i \rightarrow \frac{n}{2}$ do
        if $A[i] == A[A.length - i]$ then
            $B$.append($A[i]$);
        end
    end
    return check($B$);
end
```