Question 4 by Dan Nguyen (z5206032)

There is an array, A, of length n, where $0 < A[i] < 2^m$. Note that $i \in \mathbb{Z}[1, n]$.

Define *commonality* as the bitwise AND operation of any two non-negative integers which is not equal to zero i.e. any two non-negative integers have *commonality* if the two integers have at least 1 "1" bit in common.

Define *ordered commonality* to be the consecutive appearance of *commonality* of integers in an array without the integers necessarily being adjacent.

Define $Q(i)$ as the problem of determining the maximum number of *ordererd commonalities* in $A[1..i]$ ending with $A[i]$.

Define $\text{opt}(i)$ as the solution to $Q(i)$.

For each $1 \leq i < n$, solve for $Q(i)$ using dynamic programming where the recurrence is:

$$\text{opt}(i) = \max\{\text{opt}(j) \mid j < i,\ A[i] \,\&\, A[j] \neq 0\} + 1$$

The base case is $\text{opt}(1) = 0$.

The overall solution has a worst time complexity is $O(n^2)$. The time complexity of solving each subproblem is at worst $O(m)$ as there are at most m bits to compare in a bitwise AND operation since $A[i] < 2^m$. Therefore, recursing through A has a worst time complexity of $O(n^2 m)$.

Let there be an array, B, which is a subarray of A, and has the most *ordered commonalities* of A i.e. $B[i] \,\&\, B[i+1] \neq 0 \ \forall \ B$.

To solve for B, let there be a predecessor array, P, of length n, which stores the index i which extends the optimal solution of $Q(P[i])$ to an optimal solution of $Q(i)$. Backtracking P and inserting $A[P(i)]$ into B yields a subarray of maximum length which satisfies $Q(i)$. This has a time complexity of $O(n)$.

The optimal solution for B satisfies:

$$\max\{\text{opt}(i) \mid 1 \leq i \leq n\}$$

Therefore, the overall worst time complexity is $O(n^2 m) + O(n) = O(n^2 m)$. But the expected time complexity is $O(nm)$. :)