Question 2 by Dan Nguyen (z5206032)

Given an array, A, where $A[i] > 0 \ \forall \ i \in [1, n]$.

Let there be a subset, S, of A whose sum is $0 \leq s \leq m$.

Define $Q(s, i)$ as the problem of finding the largest s, of S of $A[1..i]$ ending with $A[i]$.

Define $\mathrm{opt}(s, i)$ as the solution to $Q(s, i)$.

For each $1 \leq i \leq n$ and initially $s = 0$, solve for $Q(s, i)$ using dynamic programming where the recurrence is:

$$\mathrm{opt}(s, i) = \max\{\mathrm{opt}(s, i-1),\ \mathrm{opt}(s - A[i], i-1) + s\}$$

The base case is $\mathrm{opt}(s, 1) = A[1]$ and $\mathrm{opt}(0, i) = 0$. The order of solving Q is important i.e. subproblems with lesser s then lesser i are solved first.

The final answer is:

$$\mathrm{opt}(m, n) = \max\{\mathrm{opt}(m, n-1),\ \mathrm{opt}(m - A[n], n-1) + m\}$$

The overall solution is $\mathrm{opt}(m, n)$. There are nm many subproblems which are solved in $O(nm)$. The time complexity of solving each subproblem is constant. Therefore, the overall time complexity is $O(nm)$ as required.