Code Jam 2018 - Round 1C

Analysis: Ant Stack

Test set 1

To solve this test set, we can use dynamic programming (DP). We define a function f(x, y) as the maximum number of ants that can form a stack (following the stack requirements given in the problem statement), in which we only consider ants from the first ant to the x-th ant, inclusive, and the sum of the ants' weight is not more than y.

We can compute the value of f(x, y) by considering two cases:

- Suppose we don't put the x-th ant on the bottom of the stack. Then we can ignore the x-th ant, and so the value of f(x, y) from this case becomes f(x - 1, y).
- Suppose we put the x-th ant on the bottom of the stack. Then we need to maximize the number of ants to be put on top of the x-th ant, which is the value of $f(x - 1, \min(6\mathbf{W_x}, y - \mathbf{W_x}))$. Counting the x-th ant as well, the value of f(x, y) from this case becomes $f(x - 1, \min(6\mathbf{W_x}, y - \mathbf{W_x})) + 1$. Note that we only consider this case if $y \geq \mathbf{W_x}$.

Between the two cases (or only one case if $y < \mathbf{W_x}$), we take the larger value as the value of f(x, y).

The answer for the problem is the value of $f(\mathbf{N}, \infty)$. Since there are $O(\mathbf{N})$ possible values for x, $O(\max(\mathbf{W}))$ possible values for y, and O(1) iterations for each computation of f(x, y), this solution runs in $O(\mathbf{N} \times \max(\mathbf{W}))$ time.

There is also another solution involving another DP formulation f', where f'(x, y) only considers ants from the x-th ant to the **N**-th ant (instead of the first ant to the x-th ant).

Test set 2

To solve this test set, we need to find the value of K, the maximum possible answer to the problem. To have a stack with as many ants as possible, where the upper-bound of the ants' weight is fixed by a constant (i.e. 10^9 in this problem), we greedily put the lightest ant possible on the bottom of the stack. In other words, a stack with as many ants as possible will have ants with weights something like (1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3, 4, 4, 5, 6, ...). We stop as soon as we need to add an ant with a weight more than 10^9. By creating a simple script, we can determine that the value of K is 139, much smaller than **N**.

Therefore, we can solve this test set with another DP formula. We define a function g(x, y) as the minimum sum of the weights of the ants that can form a stack of y ants where only the ants from the first ant to the x-th ant are considered, or ∞ if no such stack exists.

Again, we can compute the value of g(x, y) by considering two cases:

- Suppose we don't put the x-th ant on the bottom of the stack. Then we can ignore the x-th ant, and so the value of g(x, y) from this case becomes g(x - 1, y).
- Suppose we want to put the x-th ant on the bottom of the stack. We first need to check whether this is possible. We can compute the value of g(x - 1, y - 1), the minimum sum of the weights of the ants that can form a stack of y - 1 ants where only the ants from the first ant to the (x - 1)-th ant are considered. If $g(x - 1, y - 1) \leq 6\mathbf{W_x}$, then it is possible to put the

x-th ant on the bottom of the stack. The value of g(x, y) from this case becomes g(x - 1, y - 1) + $\mathbf{W_x}$.

Between the two cases (or only one case if g(x - 1, y - 1) > 6$\mathbf{W_x}$), we take the smaller value as the value of g(x, y).

The answer for the problem is the largest possible value S where g(**N**, S) < ∞. Since there are O(**N**) possible values for x, O(K) possible values for y, and O(1) iterations for each computation of g(x, y), this solution runs in O(**N**K) time.