(a) The solution for some input N is either TRUE or FALSE.

- If the solution for N is TRUE, then it is possible to purchase $N-3$ nuggets *or* $N-10$ nuggets *or* $N-25$ nuggets because we must purchase at least one box (where each "*or*" is inclusive, in other words, more than one possibility could hold).
- If the solution for N is FALSE, then it is impossible to purchase $N-3$ nuggets *and* $N-10$ nuggets *and* $N-25$ nuggets (none of the possibilities can hold).

(b) Let $A[n]=$ TRUE if it is possible to purchase exactly n nuggets (FALSE otherwise), for $-24 \leqslant n \leqslant N$.

(The negative values are not necessary but they remove the need for multiple conditions in the recurrence relation—and corresponding **if**-statements in the algorithm.)

(c)
- $A[-24]=A[-23]=\cdots=A[-1]=$ FALSE (impossible to purchase a negative number of nuggets);
- $A[0]=$ TRUE (degenerate case: buy nothing, get zero nuggets);
- $A[n]=A[n-3]\vee A[n-10]\vee A[n-25]$, for $n=1,2,\ldots,N$ (purchasing n nuggets requires purchasing a box of size 3, 10 or 25).

(d) **Algoritm:**

```
TofuNuggets(N):
    for n ← −24, −23, ..., −1:
        A[n] ← False
    A[0] ← True
    for n ← 1, 2, ..., N:
        A[n] ← A[n − 3] ∨ A[n − 10] ∨ A[n − 25]
    return A[N]
```

Runtime: $\Theta(N+25)$.

This is **not** polytime: it is an exponential function of $\log_2 N$, the *size* of input N.