An algorithm for subset sum:

The input, and array $A[1,\ldots,n]$ of numbers and a number S. The question is: can we find a subset of the elements that sums to S. Namely, a solution Sol is part of the numbers so that $\sum_{A[i]\in Sol} A[i] = S$.

We will solve actually “many” problems. For every $0 \le x \le S$, we will ask: is there a sub-array of A that sums to x? More than that, we will have a problem $P_{i,x}$ for every $0 \le i \le n$. The $P_{i,x}$ problem asks: can we choose a sub-array of $A[1,\ldots,i]$ (the first i elements) whose sum is x?

We keep a boolean matrix $p[i,x]$ for $0 \le i \le n$ and $0 \le x \le S$. The $p[i,x]$ entry will (at the end of the run of the algorithm) contain the solution (T or F) for $P_{i,x}$. For $i = 0$, namely, the empty array, we get $p[0,0] = T$ (the emptyset gives zero sum) and $p[0,x] = F$, for all $x \neq 0$.

The method: We consider $p[i,x]$ as follows. If there is a sub-array Sol that sums to x, either $A[i] \in Sol$ or $A[i] \notin Sol$.

If $A[i] \notin Sol$, then $p[i,x] = p[i-1,x]$ (as $A[i]$ does not belong to the solution). So, if $p[i-1,x]$ is T so is $p[i,x]$.

Otherwise, $A[i] \in Sol$. The other elements chosen from Sol must sum to $x - A[i]$ (so that with $A[i]$ the sum is x). Thus if $p[i-1, x-A[i]]$ is T, so is $p[i,x]$.

Subset-sum(A[1,. . . ,n],S)
$p[0,0] \leftarrow T$. For $x \leftarrow 1$ to S, $p[0,x] \leftarrow F$.
For $i = 1$ to n **Do:**
  For $x \leftarrow 0$ to S **Do:**
    If $x < 0$ then $p[i,x] \leftarrow F$.
   Else,
   $p[i,x] \leftarrow p[i-1,x] \; V \; p[i-1, x-A[i]]$

The running time is $O(S \cdot n)$.

An example:

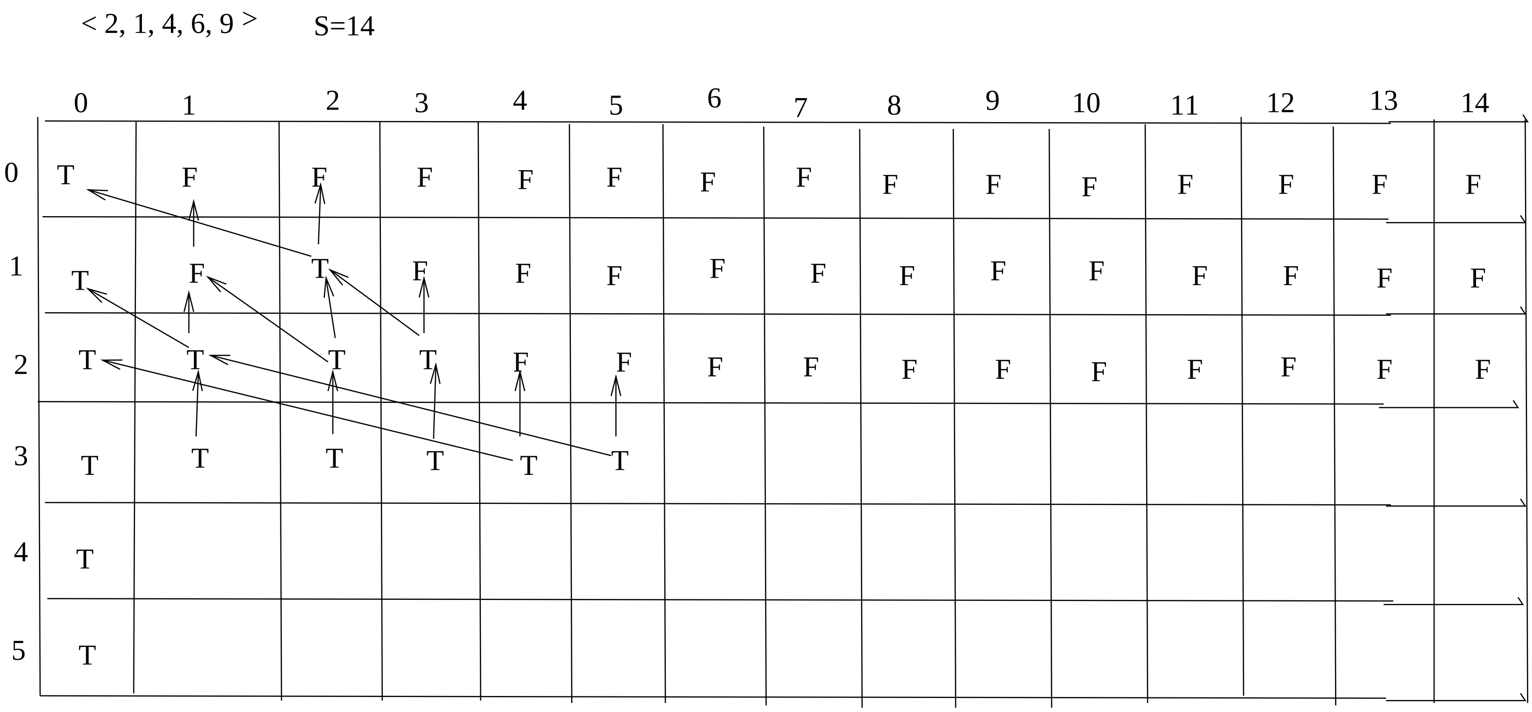

Figure 1: An example for subset sum