ECS 122A: Algorithm Design and Analysis Handout 2
UC Davis — Charles Martel January. 26, 2009

Subset Sum Handout

The subset sum problem takes as input a set of integers $L = \{x_1, \ldots, x_n\}$ and a target sum b and asks for a subset $S \subseteq L$ such that the sum of the items in S is at most b but as close to b as possible.

We can solve this by dynamic programming where we let $T[i, j] = 1$ if a subset of $x_1, \ldots, x_i$ can sum to exactly j, otherwise $T[i, j] = 0$. To fill it in row by row we do the following (where T has n rows and $b + 1$ columns).

To initialize:
$T[1, x_1] = 1$; rest of row 1 is all zeros
$T[i, 0] = 1$ for $i = 1, 2, \ldots n$

```
for $i \leftarrow 2$ to $n$
    for $j \leftarrow 1$ to $x_i - 1$
        $T[i, j] \leftarrow T[i-1, j]$
    for $j \leftarrow x_i$ to $b$
        $T[i, j] \leftarrow 1$ iff $T[i-1, j] = 1$ or $T[i-1, j-x_i] = 1$
            $T[i, j] \leftarrow 0$ otherwise
```

Thus we easily fill in the table in $\Theta(nb)$ time: this many entries, each taking O(1) time.