

Suppose that $\text{SSD}(S, t)$ is an algorithm that solves the SUBSETSUM decision problem in polytime.

Algorithm: The following algorithm solves the SUBSETSUM search problem.

```
SSS( $S, t$ ):  
  if not  $\text{SSD}(S, t)$ : return NIL  
  # Loop Invariant:  $\text{SSD}(S, t) = \text{TRUE}$ .  
  for each  $x \in S$ :  
    if  $\text{SSD}(S - \{x\}, t)$ :  
       $S \leftarrow S - \{x\}$   
  return  $S$ 
```

Correctness: If S contains no subset with sum t , the algorithm returns NIL. Else, $\text{SSD}(S, t)$ is a loop invariant so the final value of S contains some subset whose sum is exactly t . At the same time, every member of S that is not required will be eliminated during the loop. So the final value of S is the subset we are looking for.

Runtime: SSS makes $n + 1$ calls to SSD (where $n = |S|$) and uses a polynomial amount of time to create each of the sets $S - \{x\}$, so its running time is polynomial if SSD takes polytime.