Backtracking

1. Given an array $X[1..n]$ of positive integers and a nonnegative integer target T, count the subsets of positions whose entries sum to T. Two subsets are different if they use different positions, even if the values at those positions are equal.

   (a) For $X = [1, 2, 3, 4]$ and $T = 5$, list every solution. Divide your list into solutions that include the final entry and solutions that exclude it.

   (b) Suppose we have already decided which entries after position i to use. What information about those decisions do we need to count possible completions? Explain why we do not need the entire list of entries already chosen.

   (c) Let $C(i, t)$ count subsets of positions in $X[1..i]$ whose entries sum to t. Determine $C(0, 0)$, $C(0, t)$ for $t > 0$, and $C(i, t)$ for $t < 0$. What is $C(i, 0)$, and where does the assumption that all entries are positive matter?

   (d) Partition the solutions counted by $C(i, t)$ according to whether they include position i. Write a recurrence for $C(i, t)$. Why should the two recursive answers be added, rather than combined with logical OR or maximum?

   (e) Translate your recurrence into recursive pseudocode. State the initial call. Explain why every valid subset is counted exactly once.