CS207 Design and Analysis of Algorithms

Sajith Gopalan

Indian Institute of Technology Guwahati

sajith@iitg.ac.in

March 24, 2022

Backtracking

Backtracking

- Useful for constraint satisfaction problems (CSPs)
- (A CSP has a set of variables, each with a domain of values, and a set of constaints on these variables specified through relations on them. We need to find values for all variables, so that all constraints are satisfied. E.g., Sudoku)
- Incrementally builds candidates to the solutions
- Abandons a candidate ("backtracks") as soon as it becomes clear that the candidate cannot be completed into a valid solution

The Sum of Subset Problem

- Given is a set $S = \{w_1, \ldots, w_n\}$ of positive integers and a positive integer m
- Assume that the sequence $w_1, \ldots, w_n$ is in **nondecreasing** order
- Required to find every subset of S the members of which sum to m
- Assume that $w_1 \leq m$; otherwise, there is no solution
- Assume that $\sum_{i=1}^{n} w_i \geq m$; otherwise, there is no solution
- Example: $S = \{3, 8, 11, 19\}$ and $m = 30$.
- $\{3, 8, 19\}$ and $\{11, 19\}$ are the solutions
- If subsets are represented using 4-bit vectors, then the two solutions correspond to the bit vectors 1101 and 0011. 1101 means, pick the first second and fourth elements from the set. 0011 means, pick the third and fourth elements from the set.

The Sum of Subset Problem

- Let x be the bit vector that denotes the solution on the make
- $x_i = 1$ if and only if the i-th element of the set belongs to the solution
- Suppose $x_1, \ldots, x_{k-1}$ have already been fixed
- We want to print all the solutions with $x_1, \ldots, x_{k-1}$ as a prefix
- Let $sum = (\sum_{i=1}^{k-1} x_i * w_i)$ be the sum of all the elements already picked into the solution
- Let $remain = (\sum_{i=k}^{n} w_i)$ be the sum of all the remaining elements
- Since we have proceeded here, where picking w_k is a possibility, $sum + w_k \leq m$

The Sum of Subset Problem

- Suppose we choose to pick the k-th element w_k; that is, we let $x_k = 1$
- If $sum + w_k = m$, then we have found a solution:
  $x_1, \ldots, x_{k-1}, 1, 0, \ldots, 0$;
  $x_1, \ldots, x_{k-1}, 1$ is not extendible in any other way
- If $sum + w_k + w_{k+1} \leq m$, then, $x_1, \ldots, x_{k-1}, 1$ may be extendible, hence we can proceed to the next level of recursion to do just that
- If $sum + w_k + w_{k+1} > m$, then, since the w-s are in nondecreasing order, $x_1, \ldots, x_{k-1}, 1$ is not extendible

The Sum of Subset Problem

- Suppose we choose NOT to pick the k-th element w_k; that is, we let $x_k = 0$
- If $sum + (\sum_{i=k+1}^{n} w_i) \geq m$ AND $sum + w_{k+1} \leq m$, then, $x_1, \ldots, x_{k-1}, 0$ may be extendible, hence we can proceed to the next level of recursion to do just that
- (Note that $sum + (\sum_{i=k+1}^{n} w_i) = sum + remain - w_k$)
- If $sum + (\sum_{i=k+1}^{n} w_i) < m$, then $x_1, \ldots, x_{k-1}, 0$ is not extendible
- If $sum + w_{k+1} > m$, then again, since the w-s are in nondecreasing order, $x_1, \ldots, x_{k-1}, 0$ is not extendible

The Sum of Subset Problem: Algorithm

Algorithm 1 SumOfSubset($sum, k, remain$)

$x_k = 1$
if $sum + w_k = m$ **then**
    print $(x_1, \ldots, x_k)$;
else if $sum + w_k + w_{k+1} \leq m$ **then**
    SumOfSubset($sum + w_k, k + 1, remain - w_k$)
end if
$x_k = 0$
if $(sum + w_{k+1} \leq m) \wedge (sum + remain - w_k \geq m)$ **then**
    SumOfSubset($sum, k + 1, remain - w_k$)
end if
return

The Sum of Subset Problem: Algorithm

To solve the original instance, invoke SumOfSubset(0,1, $\sum_{i=1}^{n} w_i$)

The tree perspective

The tree perspective in this algorithm is simpler. The tree has 2^n leaves corresponding to the n-bit binary strings. The tree is a full binary tree of height n.
The root has two children that correspond to $x_1 = 1$ and $x_1 = 0$.

Ref: Computer Algorithms. Horowitz, Sahni, Rajasekaran. 1998.