Partially Ordered Bounded Integer Partition and Computing its Properties with Memoization Technique

A. C. Martini, J. I. Lee, and B. Simnica
Pace University, NY, NY, USA
{am78365p, jl89582p, bs87507p}@pace.edu

Partitioning a positive integer $n$ into $k$ parts is an important combinatorial problem, i.e., distributing $n$ unlabeled balls into $k$ unlabeled urns [1]. The bounded integer partition problem is counting the number of partitions where the capacity of each urn is limited to $b$. The partially ordered relation on integer partition is a level graph where $k$th level has $p_b(n,k)$ number of nodes. Edges from $k$ to $k+1$ level indicate splitting an urn to two and those from $k+1$ to $k$ level mean merging two urns into one as indicated in Fig. 1.

Let $P_{l,u}(n,k)$ be the partition of an integer $n$ into $k$ smaller integer part where each part has lower and upper bounds. Let $p_{l,u}(n,k)$ be the number of nodes in $k$ level in the partially ordered relation and its formula is given in (1).

$$p_{a,b}(n,k) = \begin{cases} 
0 & \text{if } u > l, k < 1, k > n, n < lk, \text{ or } n > uk \\
1 & \text{if } n = lk \text{ or } n = uk \\
p_{a,b}(n-1,k) + p_{l,u-1}(n-k,k) & \text{if } 1 < k < n \text{ and } lk < n < uk 
\end{cases}$$

Figure 1: Partially ordered upper bounded Integer Partition relations

This bound does not in essence change in the computational time complexity of most naïve algorithms. We found that a memoization technique makes the computation faster with the addition of this extra bound, i.e., memoization pulls even farther ahead of dynamic programming. This extra bound causes more nodes to terminate meaning memoization has even less work to do to complete the problem whereas dynamic programming would still approach the problem in the same way, not saving much time at all [1].

References