CCC '15 J5 - π-day (Editorial)

10 - 15 Marks:

For partial marks, we can treat this problem as a problem with recursive structure. We must convert the problem statement into an equation. Let $f(n, k, m)$ be the number of ways we can distribute n pieces of pie to k people. We must maintain a variable m (minimum), because a person must not get a fewer number of pieces than a person before him. Thus, for any given (n, k, m) the range of the number of pieces we can distribute lies within $[m \ldots \frac{n}{k}]$. We are dividing k because each person must at least get the number of pieces as the person in front of them. From the mentioned observations we can build a recurrence of $f(n, k, m)$.

$$f(n, k, m) = \sum_{i=m}^{|n/k|} f(n - i, k - 1, i)$$

For every i from $[m \ldots \frac{n}{k}]$, we can remove it from n resulting in $n - i$ pies, and $k - 1$ people and i becoming the new minimum.

Our recursive structure will look like so:

$$f(n, k, m) = \begin{bmatrix} 1 & k = 1 \\ 1 & n = k \\ \sum_{i=m}^{|n/k|} f(n - i, k - 1, i) & n > k \end{bmatrix}$$

However a direct recursive implementation **TLE**'s.

15 - 15 Marks:

To optimize our program, we must consider repeated operations. If we are ever calling the function with parameters (n, k, m) with it being calculated before, we can simply returned the memoized result of $f(n, k, m)$.