Homework 12 sample solution

Due 10/14/16

October 11, 2016

Stirling Numbers of the Second Kind, represented as $S(n,k)$, count (among other things) the number of ways to divide n objects into exactly k nonempty piles. For example, there are 7 ways to divide four objects into two piles:

$$\begin{array}{cc}
\{1,2,3\} & \{4\} \\
\{1,2,4\} & \{3\} \\
\{1,3,4\} & \{2\} \\
\{2,3,4\} & \{1\} \\
\{1,2\} & \{3,4\} \\
\{1,3\} & \{2,4\} \\
\{1,4\} & \{2,3\}
\end{array}$$

Stirling Numbers of the Second Kind can be calculated using the following recurrence relation:

$$S(n,k) = \begin{cases} 1, & \text{if } k = 1 \text{ or } k = n \\ kS(n-1,k) + S(n-1,k-1), & \text{otherwise} \end{cases}$$

Use this recurrence to answer the questions below. If you are interested why this recurrence is true, you may see the footnote[1] below; *however*, you do not need to understand why this works in order to solve the problem.

1. Describe a naïve recursive algorithm that computes $S(n,k)$ using this recurrence.

   See below

2. What map implementation would you use for a memoized dynamic programming solution to this problem and why?

   2D array (direct map), because the recursive function has two arguments whose values change

[1] When adding one element to go from $n-1$ to n, the new element can either go in a pile by itself or be added to an existing pile. There are $S(n-1,k-1)$ ways to put these $n-1$ elements into $k-1$ piles, and we can create the k^{th} pile by putting the new element by itself. On the other hand, there are $S(n-1,k)$ ways to put the $n-1$ elements into k piles, and we can add the new element to any of the k piles. There's only one way to put everything into one pile, and the only way to put n elements into n piles is to put everything in its own pile.

3. Describe a memoized dynamic programming solution to this problem.

   Naïve algorithm
   Memoized algorithm

```
Input: $n$: number of objects
Input: $k$: number of piles
Output: $S(n, k)$: number of ways to divide $n$ objects into $k$ nonempty
        piles
Algorithm: MemoStirling2
if $S[n, k] > 0$ then
    return $S[n, k]$
elseif $k = 1$ or $k = n$ then
    $S[n, k] = 1$
    return 1
else
    $S[n, k] = k \cdot \text{MemoStirling2}(n-1, k) + \text{MemoStirling2}(n-1, k-1)$
    return $S[n, k]$
    return $k \cdot \text{Stirling2}(n-1,\ k)\ +\ \text{Stirling2}(n-1,\ k-1)$
end
```

```
Input: $n$: number of objects
Input: $k$: number of piles
Output: $S(n, k)$: number of ways to divide $n$ objects into $k$ nonempty
        piles
Algorithm: Stirling2DynProg
$S = \text{Array}(n, k)$
Initialize $S$ to 0
return MemoStirling2($n$, $k$)
```