Final Project Documentation

Nashir Janmohamed

Due 11:59 pm, Monday, May 25

1 Recurrence Relations/Dynamic Programming

1.1 Bell numbers

The Bell numbers represent the number of ways to count partitions of (or equivalently equivalence relations on) an n element set. The n-th bell number is given by the recurrence

$$B_n = \sum_{k=1}^{n} \binom{n-1}{k-1} B_{n-k}$$

for $n \geq 0$.

1.2 Catalan numbers

The Catalan numbers form a sequence of natural numbers that occur in various counting problems, often involving recursively-defined objects. They can be expressed by the recurrence relation

$$C_{n+1} = \sum_{i=0}^{n} C_i C_{n-1}$$

for $n \geq 0$.
Their closed form is given by

$$\binom{2n}{n} - \binom{2n}{n+1}$$

1.3 Fibonacci numbers

The Fibonacci numbers, commonly denoted F_n, form a sequence such that each number is the sum of the two preceding ones, with $F_0 = 0$, $F_1 = 1$, and the recurrence given by

$$F_n = F_{n-1} + F_{n-2}$$

for $n > 1$.

1.4 Stirling numbers of the first kind

1.5 Stirling numbers of the second kind

2 Permutations and Combinations

2.1 Combinations without repetition

A combination without repetition is a selection of items from a collection, such that the order of selection does not matter. A k-combination of an n element set S is a subset of k distinct elements. The number of k-combinations is equal to the binomial coefficient given by

$$\binom{n}{k} = \frac{n!}{(n-k)!\,k!}$$

2.2 Permutations without repetition

k-permutations of n are the different ordered arrangements of a k-element subset of an n-set. This number is given by

$$P(n,k) = n \cdot (n-1) \cdot (n-2) \cdot \ldots (n-k+1)$$

2.3 Combinations without repetition

2.4 Permutations with repetition

Permutations with repetition are ordered arrangements of k elements from a set S with n elements where repetition is allowed. The number of permutations with repetition of size k is simply k^n.

2.5 Generate permutations of a string

2.6 Generate all bit strings of length n

3 Relations

3.1 # of relations

3.2 # of transitive relations

There is no known closed formula for counting the number of transitive relations. The (perhaps inefficient) approach taken in this algorithm is as follows

(1) Generate all possible relations for an n element set (given by the power set of the cartesian product of the set $\{1, 2, 3, \ldots, n\}$)

(2) For each relation generated in (1), check that for each (a, b), if there is a point of the form (b, c), then (a, c) must be in the relation

3.3 # of (ir)reflexive relations

3.4 # of symmetric relations

3.5 # of antisymmetric relations

3.6 # of equivalence relations

4 Sets

4.1 Generate power set

4.2 Generate cartesian product

5 Isomorphisms

maybe total orders?

6 Default

No documentation provided.