Combinatorics

CS 491 CAP

Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science

Spring, 2023

Objectives

- Determine the next lexicographic permutation of an array
- Calculate and use Binomial Coefficients

Permutations

- A *permutation* is a rearrangement of elements of an array.
  - Some permutations of 1,2,3,4,5:

$$\begin{array}{ccccc}
1 & 4 & 3 & 5 & 2 \\
4 & 1 & 2 & 3 & 5 \\
5 & 4 & 3 & 1 & 2 \\
3 & 2 & 5 & 1 & 2
\end{array}$$

- There are $n!$ permutations of n distinct elements.

Permutations with Repetitions

- Suppose there are repeated elements
  - n total elements,
  - n_1 elements of class 1,
  - n_2 elements of class 2, etc…
  - n_j elements of class j.

There are $\frac{n!}{n_1!n_2!\cdots n_j}$ total permutations.

- E.g., How many ways are there to line up 6 red balls and 3 white balls?
  $= \frac{9!}{6!3!}$

Calculating Next Permutations

- C++ has a `next_permutation` function, but suppose you need to do this yourself?
  - Find the highest index i such that $a[i] < a[i+1]$ This is the *pivot*.
  - Find the highest index j such that $a[j] > a[i]$.

$1 \quad 4 \quad 3 \quad 5 \quad 2$

  - In the above array, $a[i] = 3, a[j] = 5$.
  - Swap $a[j]$ and $a[i]$.

$1 \quad 4 \quad 5 \quad 3 \quad 2$

  - Then sort the following elements.

$1 \quad 4 \quad 5 \quad 2 \quad 3$

Code

```
void nextPermutation(int arr[], int n) {
    int i = n - 2;
    // Find the index of the first element that is smaller
    while (i >= 0 && arr[i] >= arr[i + 1]) i--;
    // If there is no such element, the array is already i
    if (i < 0) {
        reverse(arr, 0, n - 1);
        return;
    }
    int j = n - 1;
    // Find the index of the smallest element to the right
    while (j >= 0 && arr[j] <= arr[i]) j--;
    swap(arr[i], arr[j]);
    // Reverse the elements to the right of i
    reverse(arr, i + 1, n - 1);
}
```

Derangements

- A derangement is a permuation in which every element is relocated.
- Written $!n$

$$\begin{aligned} !0 &= 0 \\ !1 &= 0 \\ !n &= (n-1) * (!(n-1) + !(n-2)) \end{aligned}$$

- $!2 = 1, !3 = 2, !4 = 9, !5 = 44, !6 = 265, …$
- Not that common, but easy to code with DP.

Binomial Coefficients

- Coefficients of the expansion of $(x+y)^n$
  e.g. $(x+y)^4 = x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4$
- These are everywhere. E.g. Pascal's Triangle...

$$\begin{array}{c} 1 \\ 1\ 1 \\ 1\ 2\ 1 \\ 1\ 3\ 3\ 1 \\ 1\ 4\ 6\ 4\ 1 \end{array}$$

- Number of ways to chose k items from n objects. (k starts at 0...)

Formulae

- The formula: $C(n,k) = \frac{n!}{k!(n-k)!}$
- The recurrence: “either take or ignore an item”

$$\begin{aligned} C(n,0) = C(n,n) = & \quad 1 \\ C(n,k) = & \quad C(n-1,k-1) + C(n-1,k) \end{aligned}$$

- Use DP if you need a lot, but not all, of these numbers.