Number of n-element permutations with exactly k inversions

12

6

I am trying to efficiently solve SPOJ Problem 64: Permutations.

> Let A = [a1,a2,...,an] be a permutation of integers 1,2,...,n. A pair of indices (i,j), 1<=i<=j<=n, is an inversion of the permutation A if ai>aj. We are given integers n>0 and k>=0. What is the number of n-element permutations containing exactly k inversions?
>
> For instance, the number of 4-element permutations with exactly 1 inversion equals 3.

To make the given example easier to see, here are the three 4-element permutations with exactly 1 inversion:

```
(1, 2, 4, 3)
(1, 3, 2, 4)
(2, 1, 3, 4)
```

In the first permutation, 4 > 3 and the index of 4 is less than the index of 3. This is a single inversion. Since the permutation has exactly one inversion, it is one of the permutations that we are trying to count.

For any given sequence of n elements, the number of permutations is factorial(n). Thus if I use the brute force n^2 way of counting the number of inversions for each permutation and then checking to see if they are equal to k, the solution to this problem would have the time complexity $O(n! * n^2)$.

Previous Research

A subproblem of this problem was previously asked here on StackOverflow. An O(n log n) solution using merge sort was given which counts the number of inversions in a *single* permutation. However, if I use that solution to count the number of inversions for each permutation, I would still get a time complexity of O(n! * n log n) which is still very high in my opinion.

This exact question was also asked previously on Stack Overflow but it received no answers.

My goal is to avoid the factorial complexity that comes from iterating through all permutations. Ideally I would like a mathematical formula that yields the answer to this for any n and k but I am unsure if one even exists.

If there is no math formula to solve this (which I kind of doubt) then I have also seen people giving hints that an efficient dynamic programming solution is possible. Using DP or another approach, I would really like to formulate a solution which is more efficient than O(n! * n log n), but I am unsure of where to start.

Any hints, comments, or suggestions are welcome.

EDIT: I have answered the problem below with a DP approach to computing Mahonian numbers.

algorithm | permutation | dynamic-programming | combinatorics | discrete-mathematics

share edit | edited Oct 16 '13 at 1:23 | asked Oct 15 '13 at 3:49 Shashank 9,412 ●2 ●14 ●40

The fact that this has more than 1.4k views and only ten cumulative upvotes is staggering, excellent research on all accounts, +1 – WillBD Sep 9 '14 at 14:47

add a comment

asked 2 years ago
viewed 5024 times
active 1 year ago

3 Answers

active | oldest | votes

5

If there is a dynamic programming solution, there is probably a way to do it step by step, using the results for permutations of length n to help with the results for permutations of length n+1.

Given a permutation of length n - values 1-n, you can get a permutation of length n+1 by adding value (n+1) at n+1 possible positions. (n+1) is larger than any of 1-n so the number of inversions you create when you do this depends on where you add it - add it at the last position and you create no inversions, add it at the last but one position and you create one inversion, and so on - look back at the n=4 cases with one inversion to check this.

So if you consider one of n+1 places where you can add (n+1) if you add it at place j counting from the right so the last position as position 0 the number of permutations with K inversions this creates is the number of permutations with K-j inversions on n places.

So if at each step you count the number of permutations with K inversions for all possible K you can update the number of permutations with K inversions for length n+1 using the number of permutations with K inversions for length n.

share edit | answered Oct 15 '13 at 4:53 mcdowella 14.1k ●2 ●8 ●14

I think your answer is equivalent to computing the coefficients for the expansion of Product_{i=0..n-1} (1+x+...+x^i) using dynamic programming so that the results for nth row can be used to compute the results for n+1st row. Feel free to check out my answer if you are interested. I gave you +1. :) – Shashank Oct 16 '13 at 1:06

I haven't checked your answer but I wouldn't be at all surprised. There is a downloadable book at algo.inria.fr/flajolet/Publications/book.pdf which claims to teach you how to solve huge numbers of problems like this via generating functions, but I haven't found the time or motivation to work through it. – mcdowella Oct 16 '13 at 4:45

add a comment

6

It's one day later and I have managed to solve the problem using dynamic programming. I submitted it and my code was was accepted by SPOJ so I figure I'll share my knowledge here for anyone who is interested in the future.

After looking in the Wikipedia page which discusses inversion in discrete mathematics, I found an interesting recommendation at the bottom of the page.

> Numbers of permutations of n elements with k inversions; Mahonian numbers: A008302

I clicked on the link to OEIS and it showed me an infinite sequence of integers called the Triangle of Mahonian numbers.

> 1, 1, 1, 1, 2, 2, 1, 1, 3, 5, 6, 5, 3, 1, 1, 4, 9, 15, 20, 22, 20, 15, 9, 4, 1, 1, 5, 14, 29, 49, 71, 90, 101, 101, 90, 71, 49, 29, 14, 5, 1, 1, 6, 20, 49, 98, 169, 259, 359, 455, 531, 573, 573, 531, 455, 359, 259, 169, 98, 49, 20, 6, 1 . . .

I was curious about what these numbers were since they seemed familiar to me. Then I realized that I had seen the subsequence 1, 3, 5, 6, 5, 3, 1 before. In fact, this was the answer to the problem for several pairs of (n, k), namely (4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6). I looked at what was on both sides of this subsequence and was amazed to see that it was all valid (i.e. greater than 0 permutations) answers for n < 4 and n > 4.

The formula for the sequence was given as:

> coefficients in expansion of Product_{i=0..n-1} (1+x+...+x^i)

This was easy enough for me to understand and verify. I could basically take any n and plug into the formula. Then the coefficient for the x^k term would be the answer for (n, k).

I will show an example for n = 3.

$$(x^0)(x^0 + ^1)(x^0 + x^1 + x^2)$$
$$= (1)(1 + x)(1 + x + x^2)$$
$$= (1 + x)(1 + x + x^2)$$
$$= 1 + x + x + x^2 + x^2 + x^3$$
$$= 1 + 2x + 2x^2 + x^3$$

The final expansion was $1 + 2x + 2x^2 + x^3$ and the coefficients of the x^k terms were 1, 2, 2, and 1 for k = 0, 1, 2, 3 respectively. This just happens to be all valid numbers of inversions for 3-element permutations.

1, 2, 2, 1 is the 3rd row of the Mahonian numbers when they are laid out in a table as follows:

```
1
1 1
1 2 2 1
1 3 5 6 5 3 1
etc.
```

So basically computing my answer came down to simply calculating the nth Mahonian row and taking the kth element with k starting at 0 and printing 0 if the index was out of range. This was a simple case of bottom-up dynamic programming since each ith row could be used to easily compute the i+1st row.

Given below is the Python solution I used which ran in only 0.02 seconds. The maximum time limit for this problem was 3 seconds for their given test cases and I was getting a timeout error before so I think this optimization is rather good.

```
def mahonian_row(n):
    '''Generates coefficients in expansion of
    Product_{i=0..n-1} (1+x+...+x^i)
    **Requires that n is a positive integer'''
    # Allocate space for resulting list of coefficients?
    # Initialize them all to zero?
    #max_zero_holder = [0] * int(1 + (n * 0.5) * (n - 1))

    # Current max power of x i.e. x^0, x^0 + x^1, x^0 + x^1 + x^2, etc.
    # i + 1 is current row number we are computing
    i = 1
    # Preallocate result
    # Initialize to answer for n = 1
    result = [1]
    while i < n:
        # Copy previous row of n into prev
        prev = result[:]
        # Get space to hold (i+1)st row
        result = [0] * int(1 + ((i + 1) * 0.5) * (i))
        # Initialize multiplier for this row
        m = [1] * (i + 1)
        # Multiply
        for j in range(len(m)):
            for k in range(len(prev)):
                result[k+j] += m[j] * prev[k]
        # Result now equals mahonian_row(i+1)
        # Possibly should be memoized?
        i = i + 1
    return result


def main():
    t = int(raw_input())
    for _ in xrange(t):
        n, k = (int(s) for s in raw_input().split())
        row = mahonian_row(n)
        if k < 0 or k > len(row) - 1:
            print 0
```

I have no idea of the time complexity but I am absolutely certain this code can be improved through memoization since there are 10 given test cases and the computations for previous test cases can be used to "cheat" on future test cases. I will make that optimization in the future, but hopefully this answer in its current state will help anyone attempting this problem in the future since it avoids the naive factorial-complexity approach of generating and iterating through all permutations.

share edit | edited Feb 7 '15 at 19:42 musically_ut 25.9k ●5 ●61 ●78 | answered Oct 16 '13 at 0:48 Shashank 9,412 ●2 ●14 ●40

add a comment

3

The solution needs some explanations. Lets denote number of permutations with n items having exactly k inversions by I(n, k)

Now I(n, 0) is always 1. For any n there exist one and one permutation which have 0 inversions i.e., when the sequence is increasingly sorted

Now I(0, k) is always 0 since we don't have the sequence itself

Now to find the I(n, k) lets take an example of sequence containing 4 elements {1,2,3,4}

for n = 4 below are the permutations enumerated and grouped by number of inversions

```
|___k=0___|___k=1___|___k=2___|___k=3___|___k=4___|___k=5___|___k=6___|
| 1234    | 1243    | 1342    | 1432    | 2431    | 3421    | 4321    |
|         | 1324    | 1423    | 2341    | 3241    | 4231    |         |
|         | 2134    | 2143    | 2413    | 3412    | 4312    |         |
|         |         | 2314    | 3144    | 4132    |         |         |
|         |         | 3124    | 3214    | 4213    |         |         |
|         |         |         | 4123    |         |         |         |
|         |         |         |         |         |         |         |
|I(4,0)=1 |I(4,1)=3 |I(4,2)=5 |I(4,3)=6 |I(4,4)=5 |I(4,5)=3 |I(4,6)=1 |
|         |         |         |         |         |         |         |
```

Now to find the number of permutation with n = 5 and for every possible k we can derive recurrence I(5, k) from I(4, k) by inserting the nth (largest) element(5) somewhere in each permutation in the previous permutations, so that the resulting number of inversions is k

for example I(5,4) is nothing but the number of permutations of the sequence {1,2,3,4,5} which has exactly 4 inversions each. Lets observe I(4, k) now above until column k = 4 the number of inversions are <= 4 Now lets place the element 5 as shown below

```
|___k=0___|___k=1___|___k=2___|___k=3___|___k=4___|___k=5___|___k=6___|
| |5|1234 | 1|5|243 | 13|5|42 | 143|5|2 | 2431|5| | 3421    | 4321    |
|         | 1|5|324 | 14|5|23 | 234|5|1 | 3241|5| | 4231    |         |
|         | 2|5|134 | 21|5|43 | 241|5|3 | 3412|5| | 4312    |         |
|         |         | 23|5|14 | 314|5|4 | 4132|5| |         |         |
|         |         | 31|5|24 | 321|5|4 | 4213|5| |         |         |
|         |         |         | 412|5|3 |         |         |         |
|         |         |         |         |         |         |         |
|    1    |    3    |    5    |    6    |    5    |         |         |
|         |         |         |         |         |         |         |
```

Each of the above permutation which contain 5 has exactly 4 inversions. So the total permutation with 4 inversions I(5,4) = I(4,4) + I(4,3) + I(4,2) + I(4,1) + I(4,0) = 1 + 3 + 5 + 6 + 5 = 20

Similarly for I(5,5) from I(4,k)

```
|___k=0___|___k=1___|___k=2___|___k=3___|___k=4___|___k=5___|___k=6___|
|   1234  | |5|1243 | 1|5|342 | 14|5|32 | 243|5|1 | 3421|5| | 4321    |
|         | |5|1324 | 1|5|423 | 23|5|41 | 324|5|1 | 4231|5| |         |
|         | |5|2134 | 2|5|143 | 24|5|13 | 341|5|2 | 4312|5| |         |
|         |         | 2|5|314 | 31|5|44 | 413|5|2 |         |         |
|         |         | 3|5|124 | 32|5|14 | 421|5|3 |         |         |
|         |         |         | 41|5|23 |         |         |         |
|         |         |         |         |         |         |         |
|         |    3    |    5    |    6    |    5    |    3    |         |
|         |         |         |         |         |         |         |
```

So the total permutation with 5 inversions I(5,5) = I(4,5) + I(4,4) + I(4,3) + I(4,2) + I(4,1) = 3 + 5 + 6 + 5 + 3 = 22

So `I(n, k) = sum of I(n-1, k-i) such that i < n && k-i >= 0`

Also k can go up to n*(n-1)/2 this occurs when the sequence is sorted in decreasing order
https://secweb.cs.odu.edu/~zeil/cs361/web/website/Lectures/insertion/pages/ar01s04s01.html
http://www.algorithmist.com/index.php/SPOJ_PERMUT1

```
#include <stdio.h>

int dp[100][100];

int inversions(int n, int k)
{
    if (dp[n][k] != -1) return dp[n][k];
    if (k == 0) return dp[n][k] = 1;
    if (n == 0) return dp[n][k] = 0;
    int j = 0, val = 0;
    for (j = 0; j < n && k-j >= 0; j++)
        val += inversions(n-1, k-j);
    return dp[n][k] = val;
}

int main()
{
    int t;
    scanf("%d", &t);
    while (t--) {
        int n, k, i, j;
        scanf("%d%d", &n, &k);
        for (i = 1; i <= n; i++)
            for (j = 0; j <= k; j++)
                dp[i][j] = -1;
        printf("%d\n", inversions(n, k));
    }
    return 0;
}
```

share edit | edited Sep 9 '14 at 14:45 | answered Sep 9 '14 at 14:33 Vineel Kumar Reddy Kovvuri 448 ●4 ●9

add a comment

Linked

19 calculating the number of “inversions” in a permutation
3 Count Number of Sequences
3 Number of permutation with exact k inversions

Related

263 How to generate all permutations of a list in Python
1 Reduce Permutation
81 Fast permutation -> number -> permutation mapping algorithms
4 Generating a set of permutation given a set of numbers and some conditions on the relative positions of the elements
0 Permutation of n numbers
2928 How to pair socks from a pile efficiently?
3 Number of permutation with exact k inversions
2 Number of Binary Search Trees with given right arm stretch
3 Generating all permutations of pairs excluding permutations of individual elements
3 How many permutations of N integers with K pairs of numbers that can't be adjacent?

Hot Network Questions

- How to overcome a mistake you make in your proof and regain motivation?
- How to use the UK verb "reading" to mean studying/majoring in some university subject in the present tense?
- Squaring numbers that end with the digit 5
- Loading addition org files along with default scratch buffer
- Why does the Piper Cherokee (PA-28-140) engine have such low horsepower despite the very large displacement?
- Is there a "positive" definition for irrational numbers?
- Using a potentiometer to replace an absolute optical encoder
- Is there any reason to not show users incorrectly entered passwords after a successful login?
- What's the proper choice for the block size of a sponge-based hash?
- How can I improve the sharpness for tabletop still life of photography?
- Is it possible for a large pre-electricity empire to stay in touch?
- Sum or difference of two powers of two
- How to determine whether or not a plot twist is needed?
- Do effects with the duration of one minute end at the beginning or the end of your turn?
- Script to detect if a certain website is up?
- What do you call the bunch of hair you grow long on the back of your head?
- Which is stronger, rope or rope with knots?
- Why do most people think it's a bad idea to read from slides?
- Is Expeditious Retreat essentially a spell to double your speed?
- How can cheating be prevented in a FLOSS multiplayer game?
- Meditation , Compassion. How to handle people who dominate you in office?
- Greater than less than greater than something fishy
- Are there languages in which "coffee" is not a cognate of a root containing k/q and f/h/w?
- Extend Frames On Bezier Circle Path Animation

Your Answer

Links | Images | Styling/Headers | Lists | Blockquotes | Code | HTML | advanced help »

community wiki

Post Your Answer

Not the answer you're looking for? Browse other questions tagged algorithm permutation dynamic-programming combinatorics discrete-mathematics or ask your own question.

question feed