arXiv: math.PR/00000000

Seven Competing Horses*

Zhikai Wang?? http://www.heteroclinic.net

Montreal, Quebec, Canada e-mail: wangzhikai@yahoo.com *url:*

Abstract: Seven different horses are racing. We try to find all the possible results.

Keywords and phrases: kwdseven kwdhorses kwdrace.

Contents

1. Introduction

There is a race involving seven horses. If there is not any tie, there will the total permutations of seven results. If there exists tie or ties, we have to put more consideration.

2. About Ties

If we are going to put seven horses into seven different large trailers for example. All the possibilities sum up to 7^7. But for racing, it is different. For trailer assignments, the trailers are different, case three horses in trailer one and four horses in trailer two is different from the case that three horses are in trailer five and four horses are in traler seven. For racing ranks, we only have three horses rank number one four horses ranking number two (or four), merging the two diferent cases in the trailers' assignment. So, even we can enumerate all 7^7 possibilities, the task to merge congruent cases remains big.

*This latex file is built as per latex template at http://www.e-publications.org. Any dispution upon the usage of the above template, please contact the author.

3. The Proposed Algorithm

First let's consider one horse is in the race. It will be number one.
Then we put one more horse. The new horse can be put to three positions, before, after and same position as the original horses. If there is a third horse, we find if the previous two have a tie, then it is the case like put new horse to one existing; if the previous two have rank two, then the new horse can find five poisitions, each leading to a different result.
By now we observe that the positions a new horse can find does not rely on the number of existing horses, but on the number of ranks of existing horses. If the existing ranks is m, the new horse can find $2m + 1$ new positions. If the new horse is put at an odd position, it will increase the number of ranks by one; if the new horse is put at an even position, it will not change the existing number of ranks. It is difficult to give the time and space complexity of this algorithm.

Algorithm 1 Find the total number of possibilites of n racing horses

```
input: a positive integer $n$.
output: an integer, which are the number of results of $n$ horses racing.
$i \Leftarrow 1$
let $l$ be an empty linked list of integers.
add 1 to $l$.
while $++i \leq n$ do
   Initialze $l_2$ as empty linkedlist of integers.
   for each $j$ in $l$ do
      Initialze $l_{tmp}$ as empty linkedlist of integers.
      for s be 1:2j+1 do
         if $s \mod 2 \equiv 1$ then
            append $j + 1$ to $l_{tmp}$
         else
            append $j$ to $l_{tmp}$
         end if
      end for
      append $l_{tmp}$ to $l_2$
   end for
   empty $l$ then append $l_2$ to $l$
end while
return the size of $l$
```

The only thing we can tell is there will be less than 7^7 possibilities for $n = 7$. It will be a tree structure of height n. Each none-leaf node has twice the leaf's value plus one sub-nodes.

4. Discussion

This article is not peer reviewed. Neither are the methods tested and verified by concrete computer program. Discretion advised for serious readers.

References