679. 24 Game ☐ (/problems/24-game/)

Average Rating: 2.50 (4 votes)

Sept. 16, 2017 | 6.5K views

You have 4 cards each containing a number from 1 to 9. You need to judge whether they could operated through `*`, `/`, `+`, `-`, `(`, `)` to get the value of 24.

Example 1:

```
Input: [4, 1, 8, 7]
Output: True
Explanation: (8-4) * (7-1) = 24
```

Example 2:

```
Input: [1, 2, 1, 2]
Output: False
```

Note:

1. The division operator `/` represents real division, not integer division. For example, 4 / (1 - 2/3) = 12.
2. Every operation done is between two numbers. In particular, we cannot use `-` as a unary operator. For example, with `[1, 1, 1, 1]` as input, the expression `-1 - 1 - 1 - 1` is not allowed.
3. You cannot concatenate numbers together. For example, if the input is `[1, 2, 1, 2]`, we cannot write this as 12 + 12.

Approach #1: Backtracking [Accepted]

Intuition and Algorithm

There are only 4 cards and only 4 operations that can be performed. Even when all operations do not commute, that gives us an upper bound of $12 * 6 * 2 * 4 * 4 * 4 = 9216$ possibilities, which makes it feasible to just try them all. Specifically, we choose two numbers (with order) in 12 ways and perform one of 4 operations (12 * 4). Then, with 3 remaining numbers, we choose 2 of them and perform one of 4 operations (6 * 4). Finally we have two numbers left and make a final choice of 2 * 4 possibilities.

We will perform 3 binary operations (`+`, `-`, `*`, `/` are the operations) on either our numbers or resulting numbers. Because `-` and `/` do not commute, we must be careful to consider both `a / b` and `b / a`.

For every way to remove two numbers `a`, `b` in our list, and for each possible result they can make, like `a+b`, `a/b`, etc., we will recursively solve the problem on this smaller list of numbers.

Java   Python   Copy

```java
1   class Solution {
2       public boolean judgePoint24(int[] nums) {
3           ArrayList A = new ArrayList<Double>();
4           for (int v: nums) A.add((double) v);
5           return solve(A);
6       }
7       private boolean solve(ArrayList<Double> nums) {
8           if (nums.size() == 0) return false;
9           if (nums.size() == 1) return Math.abs(nums.get(0) - 24) < 1e-6;
10
11          for (int i = 0; i < nums.size(); i++) {
12              for (int j = 0; j < nums.size(); j++) {
13                  if (i != j) {
14                      ArrayList<Double> nums2 = new ArrayList<Double>();
15                      for (int k = 0; k < nums.size(); k++) if (k != i && k != j) {
16                          nums2.add(nums.get(k));
17                      }
18                      for (int k = 0; k < 4; k++) {
19                          if (k < 2 && j > i) continue;
20                          if (k == 0) nums2.add(nums.get(i) + nums.get(j));
21                          if (k == 1) nums2.add(nums.get(i) * nums.get(j));
22                          if (k == 2) nums2.add(nums.get(i) - nums.get(j));
23                          if (k == 3) {
24                              if (nums.get(j) != 0) {
25                                  nums2.add(nums.get(i) / nums.get(j));
26                              } else {
27                                  continue;
28                              }
```
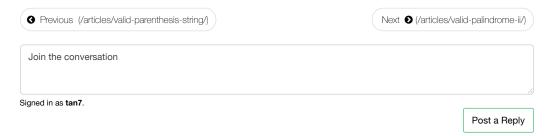
Complexity Analysis

- Time Complexity: $O(1)$. There is a hard limit of 9216 possibilities, and we do $O(1)$ work for each of them.

- Space Complexity: $O(1)$. Our intermediate arrays are at most 4 elements, and the number made is bounded by an $O(1)$ factor.

Analysis written by: @awice (https://leetcode.com/awice)

Rate this article:

(/ratings/107/247/?return=/articles/24-game/) (/ratings/107/247/?return=/articles/24-game/) (/rati

⊙ Previous (/articles/valid-parenthesis-string/)   Next ⊙ (/articles/valid-palindrome-ii/)

Join the conversation

Signed in as **tan7**.

Post a Reply

K   **kevincongcc** commented last week

Thanks you very much!!!
(https://discuss.leetcode.com/user/kevincongcc)

A   **arnie001** commented last month

What would be the time complexity for arbitrarily sized number lists (n=4 here)? Each
(https://discuss.leetcode.com/user/arnie001)
invocation of solve method does O(n^2) work and the max depth of the recursion is n. So is
O(n^3)?

rynehx commented 2 months ago

(https://discuss.leetcode.com/user/rynehx)

Question about the 6 * 4 part of the analysis. We we already have 12 possibility of the first two and four operations between them then we have 12 * 4 for first two elements. Why is it correct to do choose 2 from remaining 3? Because since we established 12 * 4 for the first two we have 2 remaining then so wouldn't we choose from remaining 2 and have 4 operations. So I thought it would be (12 * 4) * (2 * 4) * (1 * 4)

rainmaker9001 commented 3 months ago

(https://discuss.leetcode.com/user/rainmaker9001)

@smallcoderrrrrr (https://discuss.leetcode.com/uid/37787), 1e-6 is done because of floating point division rounding errors.

Also, in the Python code,
B = [A[k] for k in xrange(len(A)) if i != k != j]
is slightly slow because we have to copy an array. Of course, since this problem is for 4 cards only, this is a constant time operation. However, if we want to generalize this problem to n cards, it might make sense to use a hashmap from cards to counts rather than a Python list, and rather than making a new array we can remove 2 elements from the set and then add an element back to the set.

smallcoderrrrrr commented 3 months ago

(https://discuss.leetcode.com/user/smallcoderrrrrr)

```
'''
if (nums.size() == 1) return Math.abs(nums.get(0) - 24) < 1e-6;
'''
```

Where does 1e-6 come from?

awice commented 5 months ago

(https://discuss.leetcode.com/user/awice)

@tsar2512 (https://discuss.leetcode.com/uid/152073) It means choosing (a, b) is different from choosing (b, a). In total there are 12 ways to choose 2 items from 4 with order, as enumerated in one of my posts earlier.

tsar2512 commented 5 months ago

(https://discuss.leetcode.com/user/tsar2512)

what does chose two numbers with order mean here? I'm a little confused

CExplr commented 6 months ago

(https://discuss.leetcode.com/user/cexplr)

@cygnus (https://discuss.leetcode.com/uid/116661) in another way, it is a permutational problem. Choosing 2 numbers from 4 is 4C2 which is n!/((n-r)!) = 4!/((4-2)!) = 4*3*2/2 = 12

awice commented 6 months ago

(https://discuss.leetcode.com/user/awice)

@cygnus (https://discuss.leetcode.com/uid/116661) For example say the numbers are `(a, b, c, d)`. The 12 ways are `(a, b), (a, c), (a, d), (b, c), (b, d), (c, d)` and the other 6 reflections `(b, a), (c, a), (d, a), (c, b), (d, b), (d, c)`.

cygnus commented 6 months ago

(https://discuss.leetcode.com/user/cygnus)

"we choose two numbers (with order) in 12 ways " This is wrong, right? it should be 9 * 9 = 81.

View original thread (https://discuss.leetcode.com/topic/103917)   [Load more comments...]