808. Soup Servings

DescriptionHintsSubmissionsDiscussSolution

There are two types of soup: type A and type B. Initially we have `N` ml of each type of soup. There are four kinds of operations:

1. Serve 100 ml of soup A and 0 ml of soup B
2. Serve 75 ml of soup A and 25 ml of soup B
3. Serve 50 ml of soup A and 50 ml of soup B
4. Serve 25 ml of soup A and 75 ml of soup B

When we serve some soup, we give it to someone and we no longer have it. Each turn, we will choose from the four operations with equal probability 0.25. If the remaining volume of soup is not enough to complete the operation, we will serve as much as we can. We stop once we no longer have some quantity of both types of soup.

Note that we do not have the operation where all 100 ml's of soup B are used first.

Return the probability that soup A will be empty first, plus half the probability that A and B become empty at the same time.

Example:
Input: `N = 50`
Output: `0.625`
Explanation:
`If we choose the first two operations, A will become empty first. For the third operation, A and B will become empty at the same time. For the fourth operation, B will become empty first. So the total probability of A becoming empty first plus half the probability that A and B become empty at the same time, is 0.25 * (1 + 1 + 0.5 + 0) = 0.625.`

Notes:

- `0 <= N <= 10^9`.
- Answers within `10^-6` of the true value will be accepted as correct.

Seen this question in a real interview before?

- Difficulty:Medium
- Total Accepted:665
- Total Submissions:2.6K
- Contributor:awice

-
- Subscribe to see which companies asked this question.

  Related Topics

-

The most important hint in this problem for me is `"Answers within 10^-6 of the true value will be accepted as correct."`.
And when I get timed out or runtime error with my DP, I tried to print out results of each call for each N.

They are monotonically increasing, and getting closer and closer to 1.
If you print it out, you will observe that when it is as large as 5551 the gap between the result and 1 is less than 10^-6.

Note that, the memoization is necessary, otherwise it will get TLE.

```
double sub(int an, int bn, unordered_map<string,float> &dd)
{
    if(an<=0 && bn<=0) return 0.5;
    if(an<=0) return 1;
    if(bn<=0) return 0;
    char keys[20];
    sprintf(keys,"%d,%d",an,bn);
    string key = string(keys);
    if(dd.count(key)) return dd[key];
    double c1 = sub(an-100,bn,dd);
    double c2 = sub(an-25,bn-75,dd);
    double c3 = sub(an-50,bn-50,dd);
    double c4 = sub(an-75,bn-25,dd);
    double s = c1+c2+c3+c4;
    s*=0.25;
    dd[key]=s;
    return dd[key];
}

double soupServings(int N) {
    if(N>5551)
        return 1;
    unordered_map<string,float> dd;
    return sub(N,N,dd);
}
```