Complete Search

George Tang

9/29/17

1 Introduction

In Complete Search, up to the entire search space is searched to find the solution. Complete search solutions are simple, and should be the one of the first approaches you consider when attacking the problem, as they provide important insight about the problem. However, they are usually slow, as they enumerate all possible combinations.

Complete Search problems come in all shapes and sizes, and are very popular in contests like ACSL and ACM, less in USACO. The best way to master Complete Search is to become familiar with general approaches, which comes with doing many problems. As such, in this lecture we focus on examples and less on any specific technique.

2 Iterative Complete Search

Problem Statement: Given three integers A, B, C, ($0 \leq$ A, B, C ≤ 10000), find all positive triples of x, y, z such that x + y + z = A, x * y * z = B, and $x^2 + y^2 + z^2 = $ C.

Solution: This problem requires three for loops. The naive approach is to iterate from 0 to 10000 for x, y, z, but clearly this will result in Time Limit Exceeded (TLE). To make it run in time, we realize x, y, z cannot exceed 100 as that would violate the third condition. Now we have a O(N^3) solution where N = 100, which will run in time.

Algorithm 1 Numbers

```
for x in [0, 100] do
    for y in [0, 100] do
        for z in [0, 100] do
            if x + y + z = A and x * y * z = B and x^2 + y^2 + z^2 = C then print x, y, z
```

Exercise 1: There are N ($1 \leq$ N ≤ 15) lights in a room, and each light can be turned on or off. There are M ($1 \leq$ M ≤ 50) constraints, where each has two lights that cannot be simultaneously turned on. How many valid combinations of light are possible? (Hint, each enumeration of lights can be represented as a binary number: i.e. lights 1 and 3 on with 5 total lights can be represented as 00101, and each binary number corresponds to a base 10 number.)

3 Recursive Complete Search

Problem Statement: Farmer Noah has three buckets, each with capacity of A, B, C ($1 \leq$ A, B, C ≤ 20). Initially, bucket A is always full of milk and B and C are empty. Farmer Noah can pour a bucket into another, and doesn't stop until the first bucket is empty or second is full. How many different ways can milk occupy the three buckets? A state is defined as same if the amount of milk in each bucket is the same.

Solution: We are given an initial state and the rules for transitions into other states. We start at the initial state proceed as follows: if we have not seen the state, we increment the total number of ways and recur on the states determined by the transitions; if we seen the states, do nothing and return to the previous state.

Algorithm 2 Milk

```
seen = [25][25][25]                          ▷ [a][b][c]: A has a milk, B has b milk, C has c milk
count = 0
c = [3]                                      ▷ capacities of bucket in order of A, B, C
Recur(initial)
print count

function Pour(x, y, z, m)                    ▷ return state where x was poured into y
    temp = [3]                               ▷ state with 3 variables
    temp[z] = m[z]                           ▷ bucket z is unchanged
    if m[x]+m[y] > c[y] then                 ▷ overflow
        temp[x]=m[x]+m[y]-c[y]
        temp[y]=c[y]
    else
        temp[x]=0
        temp[y]=m[x]+m[y]
    return temp

function Recur(m)                            ▷ process state m
    if seen[m[x]][m[y]][m[z]] then return
    seen[m[x]][m[y]][m[z]] = 1
    count++
    Recur(pour(0, 1, 2))
    Recur(pour(1, 0, 2))
    Recur(pour(0, 2, 1))
    Recur(pour(1, 2, 0))
    Recur(pour(2, 0, 1))
    Recur(pour(2, 1, 0))
```

4 Tips

4.1 Precalculation

Problem Statement: A bisquare is expressed of the form $p^2 + q^2$. Given N≤25, the length of the arithmetic progression to search and M≤250, the limit of p and q, find all arithmetic progressions possible.

Solution: Precompute the bisquares. For all bisquares, loop through all possible differences, which is bound. Then check to see if the bisquares from this bisquare to the Nth next bisquares forms an arithmetic progression.

4.2 Working Backwards

Problem Statement: There is a 2D grid of size s (s≤1000) of of N rats (N≤10000). Determine which (x,y) should be bombed so as many rats are killed as possible in square grid (x-d, y-d) and (x+d, y+d) (d≤10).

Solution: Instead of looping through all (x,y) and counting the rats killed, which is O(s^2 d^2), for each rat population at (x, y), add to killed[i][j] if |i-x|≤d and |j-y|≤d. This gives us O(N d^2).