Problem Set 7

Name: SID:

Spring 2016   GSI:

Warmup

1. Correcting XYZ

a. For any java program, P, define $S(P)$ to be the set of all Java programs P' that output the same result as P on all inputs. Formally, $S(P) = \{P' \mid \forall x : P(x) = P'(x),\ P'(x)\ \text{halts if and only if}\ P(x)\ \text{halts}\}$. XYZ claims that they have built an optimal java-program-shortner. Formally, they claim to have a procedure **optimalShortner** such that:

```
for every java program P:
    let P' = optimalShortner(P)
    then:
        P' is in S(P)
        for all P'' in S(P), length(P'') >= length(P')
```

Prove that XYZ is wrong.

> **Solution**

b. Having failed with source code shortening, XYZ now tries their luck with runtime optimization. XYZ claims that they have built a new optimizer **optimizer** such that:

```
for every java program P:
    let P' = optimizer(P)
    then:
        forall x:
            if P(x) halts
            then P'(x) halts within 2^{|x|} steps
            else P'(x) infinite loops
```

Namely, XYZ claims that their optimizer outputs an equivalent program such that: if $P(x)$ halts, then $P'(x)$ will halt within $2^{|x|}$ steps; if $P(x)$ does not halt, then $P'(x)$ does not halt either. Prove that **optimizer** can not exist.

> **Solution**

c. Let XYZ-phone be a smart phone with 16GB of storage, 2GB of RAM, and additional state of 1MB (i.e. CPU registers, etc).

Prove that for any program P which does not interact with the rest of the world (no user input, no network connection, no wifi, no bluetooth, no sensors), it is possible to determine whether all executions of P halts or whether some execution of P infinite loops. (Different executions of P may behave differently due to the random number generator, the initial state of the phone when P is loaded, and due to non-determinism caused by multi-threading.

> **Solution**

2. Printing all x where $M(x)$ halts

Prove that it is possible to write a program P which:

```
* takes as input M, a java program
* runs forever, and prints out strings to the console
* for every x, if M(x) halts, then P(M) eventually prints out x
* for every x, if M(x) does NOT halt, then P(M) never prints out x
```

Solution

# 3. Lexicographical output is impossible

Lexicographical ordering of strings means (1) shorter strings are in front of longer strings (2) for two strings of the same length, they are sorted in alphabetical order.

Prove that it's impossible to solve the above problem if we require the output be in lexicographical order.

Solution

4. Maze

Let's assume that Tom is located at the bottom left corner of the maze below, and Jerry is located at the top right corner. Tom of course wants to get to Jerry by the shortest path possible.
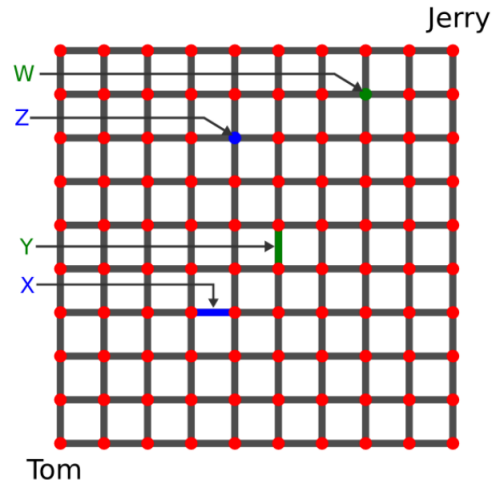
a. How many such shortest paths exist?

Solution

b. How many shortest paths pass through the edge labelled X? The edge labelled Y? Both the edges X and Y? Neither edge X nor edge Y?

Solution

c. How many shortest paths pass through the vertex labelled Z? The vertex labelled W? Both the vertices Z and W? Neither vertex Z nor vertex W?

Solution

5. Counting Practice!

The only way to learn counting is to practice, practice, practice – so here is your chance to do so. No need to justify your answers or show your calculations on this problem. We encourage you to leave your answer as an expression (rather than trying to evaluate it to get a specific number).

   a. How many 10-bit strings are there that contain exactly 4 ones?

> **Solution**

   b. How many different 13-card bridge hands are there? (A bridge hand is obtained by selecting 13 cards from a standard 52-card deck. The order of the cards in a bridge hand is irrelevant.)

> **Solution**

   c. How many different 13-card bridge hands are there that contain no aces?

> **Solution**

   d. How many different 13-card bridge hands are there that contain all four aces?

> **Solution**

   e. How many different 13-card bridge hands are there that contain exactly 6 spades?

> **Solution**

   f. How many 99-bit strings are there that contain more ones than zeros?

> **Solution**

   g. If we have a standard 52-card deck, how many ways are there to order these 52 cards?

> **Solution**

   h. Two identical decks of 52 cards are mixed together, yielding a stack of 104 cards. How many different ways are there to order this stack of 104 cards?

> **Solution**

   i. How many different anagrams of FLORIDA are there? (An anagram of FLORIDA is any reordering of the letters of FLORIDA, i.e., any string made up of the letters F, L, O, R, I, D, and A, in any order. The anagram does not have to be an English word.)

> **Solution**

   j. How many different anagrams of ALASKA are there?

> **Solution**

   k. How many different anagrams of ALABAMA are there?

> **Solution**

   l. How many different anagrams of MONTANA are there?

> **Solution**

   m. We have 9 balls, numbered 1 through 9, and 27 bins. How many different ways are there to distribute these 9 balls among the 27 bins?

n. We throw 9 identical balls into 7 bins. How many different ways are there to distribute these 9 balls among the 7 bins such that no bin is empty?

o. How many different ways are there to throw 9 identical balls into 27 bins?

p. There are exactly 132 students currently enrolled in CS70. How many different ways are there to pair up the 132 CS70 students, so that each student is paired with one other student?

q. How many ways are there to arrange n 1s and k 0s into a sequence?

r. How many solutions does
$$x_0 + x_1 + \ldots + x_k = n$$
have, if all x's must be non-negative integers?

s. How many solutions does
$$x_0 + x_1 = n$$
have, if all x's must be *strictly positive* integers?

t. How many solutions does
$$x_0 + x_1 + \ldots + x_k = n$$
have, if all x's must be *strictly positive* integers?

6. Prove the following identities by combinatorial argument:

a.
$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Solution

b.
$$\binom{2n}{2} = 2\binom{n}{2} + n^2$$

Solution

c.
$$\sum_{k=0}^{n} k\binom{n}{k} = n2^{n-1}$$

Solution

d.
$$\sum_{k=j}^{n} \binom{n}{k}\binom{k}{j} = 2^{n-j}\binom{n}{j}$$

Solution

e.
$$\sum_{i=0}^{k} \binom{m}{i}\binom{n}{k-i} = \binom{m+n}{k}$$

Solution