ChenruiXu 515Homework 3

Chenrui Xu

October 2021

1 Question 1

We can view the process into a bipartite graph. Where s (beginer) is the device and t (end) is the receiver. We can use Ford-Fulkerson method to let reciver get max flow F_r.

So each time, the leak should be $F_d - F_r$
So the total leak is $\sum_{i=1}^{C}(F_d - F_r)$ and it is supposed to be less than one.
And the average should be $avg(leak) = \frac{log(\sum_{i=1}^{C}(F_d - F_r))}{numberofC}$
as we turn the flow into information (log).

2 Question 2

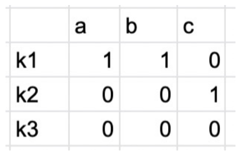
Step 1: So in this question, we can view it as a Linear programming question. There are x1, x2, to x7 and x, y, z, totally ten variables. With each one of ten lines, we can get at least one equal linear combination (like $y := 2x_1 + 3x_2 - 5$). If it is a if/else line, we can also get an inequality function (like $y > 12x_1 - z$ or $y < 12x_1 - z$). So, totally, we can get totally ten equal linear combinations and n inequality functions (n is a variable,based on how many if/else are there).

Step 2: So based on ten equal linear combinations, we only keep x, x1 to x7 eight variables and get rid of y and z by using replacing method. Also, if necessary, we can replace any x_i among x1 to x7 by using of other $x's$ linear combinations. We finally can use some or all of x1 to x7 to express x. That is

$$x = \sum_{i=1}^{7} a_i * x_i + b$$

Recall there are n inequality functions, just make sure that we replace all variables not appear in $x = \sum_{i=1}^{7} a_i * x_i + b$ with other variables.

Step 3: Note that there are 2^n possible situations given n if/else modules.



	a	b	c
k1	1	1	0
k2	0	0	1
k3	0	0	0

Figure 1:

For example:

if inequal1: statement 1 else: statement 2

Whatever situation is, we can get one inequality function ($inequal1$ or $notinequal1$) and corresponding statement. There are 2^n possible situations, which means, there are 2^n corresponding A and b in $x = \sum_{i=1}^{7} a_i * x_i + b$.

Step 4: So based on corresponding situations, we put $x = \sum_{i=1}^{7} a_i * x_i + b < 0$ (a,b) and other inequality functions into a linear inequality programming problem and solve it then. Note that if it has solution, means $x < 0$ is possible. If not (conflicts of inequalities or other situations), $x < 0$ is not possible.

3 Question 3

B_k is the number of P, which is sum of bell numbers.

$B_k = \sum_{i=0}^{n} C(n,i) * B_i$ where $C(n,i)$ the number of situations that pick i elements from n and B_i is the numbers of possible results given cardinality i. So, we can get that there are totally B_k Ps given cardinality n and we can map them from 1 to n one-on-one.

Method 1: Given set A $a, b, c...n$, backtrack to get all possible subsets, mark as A'.

Backtrack again on A' to get all possible subsets of A', and get rid of the subset if exist same element appears more than one time. We get the set mark as B.

B is the set of which elements are all possible P. As the length of set B is B_k, we map those elements with corresponding index.

Method 2: For the given set with n elements, we create matrix to represent different P. If K is $[a, b, c]$, then we can exhaustive all possible P matrix (if for all K_i(any lines in matrix) in P_i, K_i is also in P_j, get rid of P_j). Like Figure 1 (at the head of Page two), it is the set $[[a, b], [c]]$.(Make sure the sum of each colunm is 1)

After we encoding all those possible matrix, we splice K1,K2 to Kn to a binary number (That is 110001000). One P corresponding to one unique binary number. Sort them and return to the index, we get the one-on-one mapping from 1,...,B_k to set P.

4 Question 4

Since there are 2048 nodes in the graph, so let $2^k = 2048$. And the answer is k=11.

So there are total $2 * 11 = 22$ boolean variables.

5 Question 5

We turn 40 into binary format, so the store space should be $log_2 40$. The result is between 5 and 6, so we take 6 bits as the result.