CSCC63 TUT 0002
Tutorial 12

- Assignment Help (Q7 and more)
- Exam Contents/Review

What I suspect will be on the exam

Almost Guaranteed:
1 Recognizable/Co-recognizable/Both/Neither question
(could also be one simple one, and one more difficult one that requires dovetailing)
2 NP-complete questions (verifier & reduction)
Generalized geography reduction, given the TQBF, write out the GG tree

Highly Likely:
PCP/CFG reduction
Oracle TM question (given an oracle for some language, write a TM that uses the oracle to decide another)
P-Space question (write an algorithm, justify why it runs in PSPACE)

Not as Likely:
Approximation question, show that something cannot be approximated with a constant ratio
Counting Problem (#P question)

Bonus Question Candidates:
Parsimonious Reduction, PCP/CFG reduction, #P question, approximation question

Tips for the exam:

Finish all the NP Verifier Questions first, write the polytime verifiers
Do the TQBF to Generalized geography question

After, do all of the questions you already know (if there isn't any thats okay)

Prioritize recognizability ones. L = { <M> | L(M) = { 1 } }, L = { <M> | L(M) subet { 1 } },

Approximation Question,

Given a language, Show that it cannot be approximated better than a ratio of 1/4

1. Assume that such an approximation exists, that is to say a function can approximate this language with a result that is better than 1/4th.
2. Write a function using this approximation function and approximate the problem to a ratio better than 1/2.
3. Your function should look like this

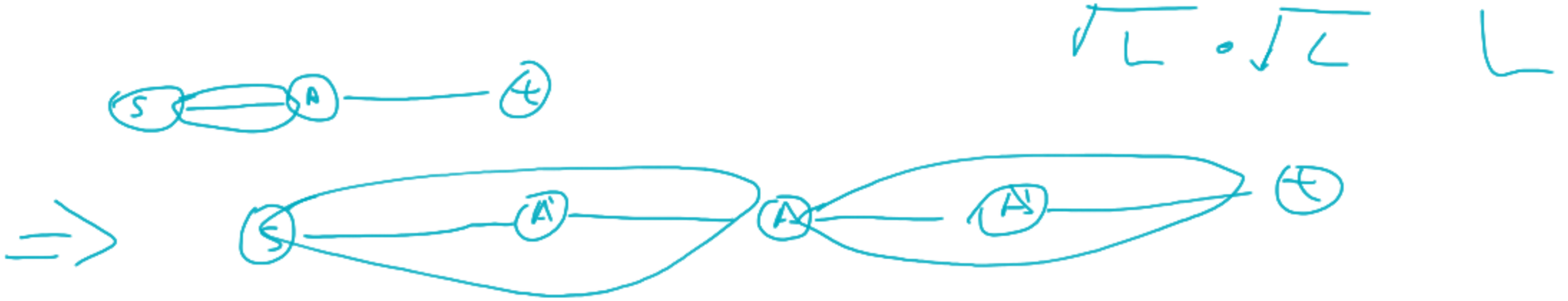
P on input <x>
  y = square x in some way
  run the approximation on y, receive a better approximation than 1/2

LONGEST

Input A direct graph G, nodes s, t

Output the length of the longest path from s to t in G

Assume that a function named approx can compute this length up to a ratio of 1/4,



When we run approx on G', any path of length l in G' can be mapped to a path of length sqrt(l) in G

P on input <G, s, t>

G' Modify(G)

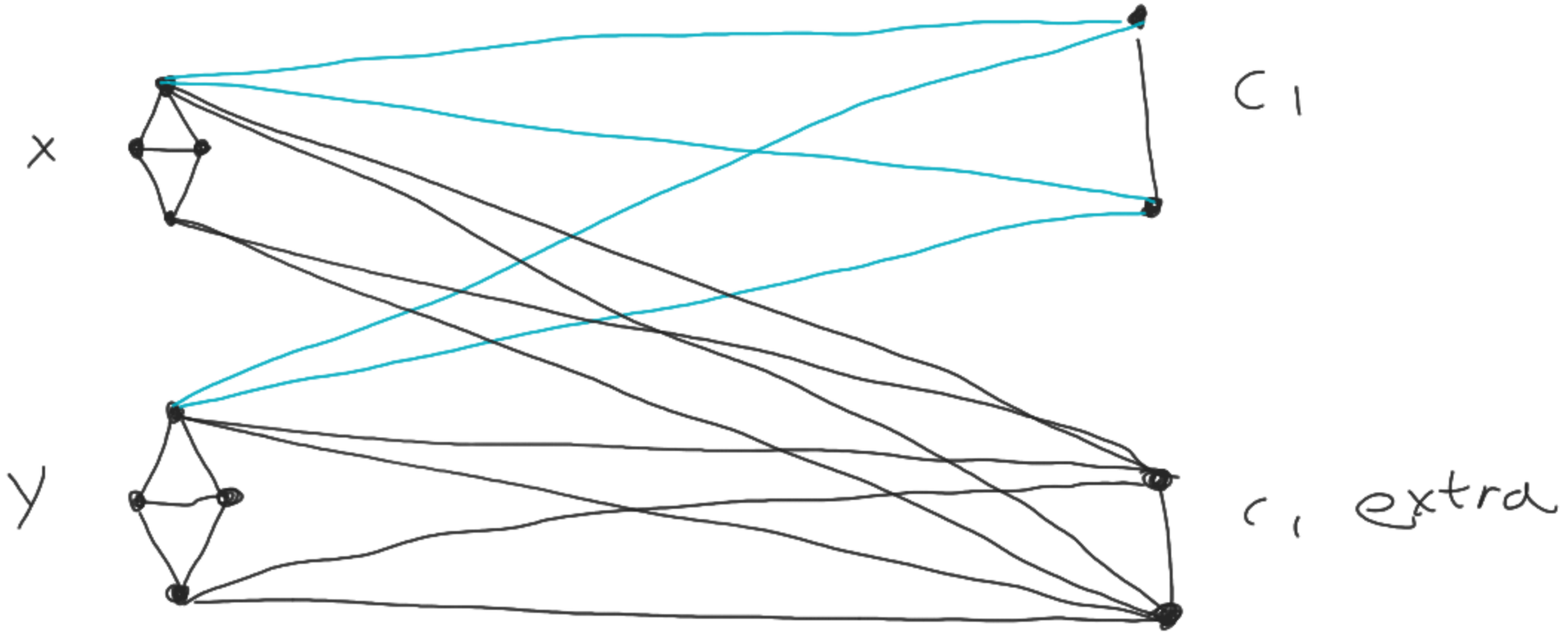
get the results of approx(G')

suppose that G has a longest path length of l, then G' has longest path l^2

the result of approx(G') gives us a path length of (l^2/4)

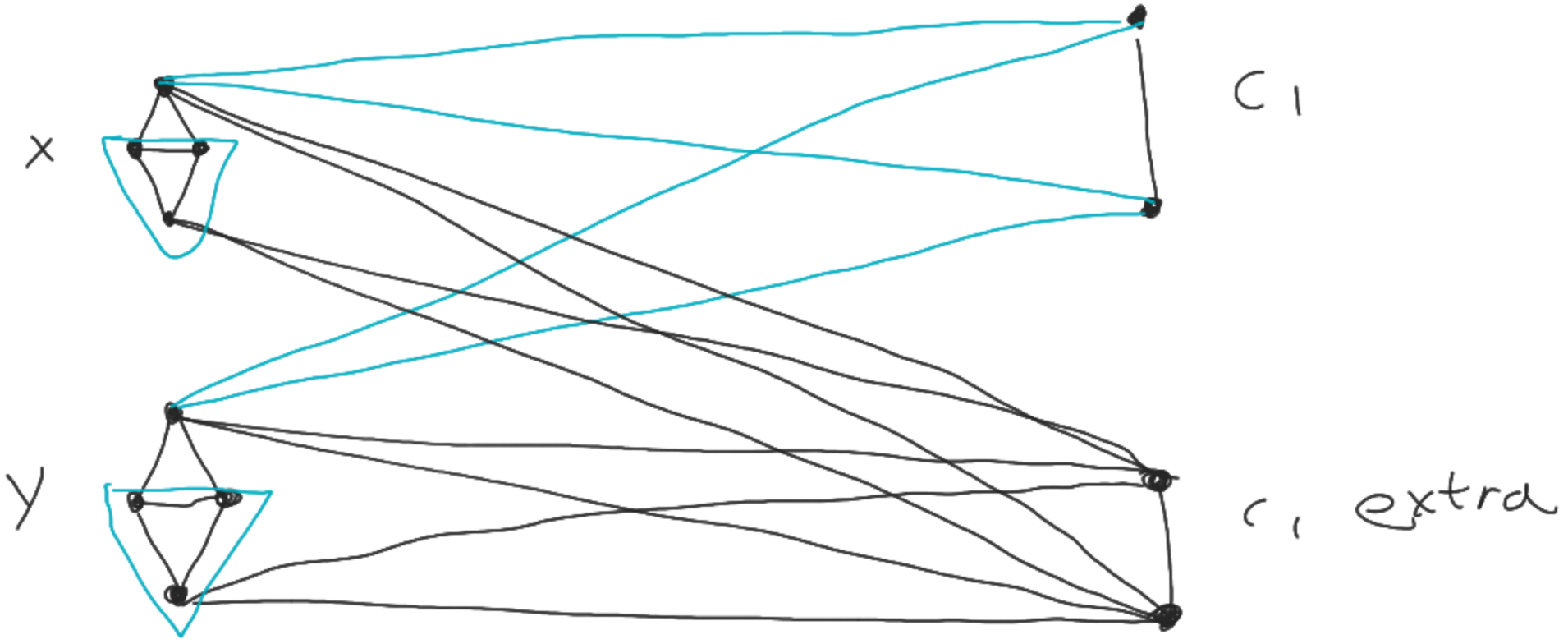
this result maps to a path length of (sqrt(l^2/4)) = l/2

(x or y)

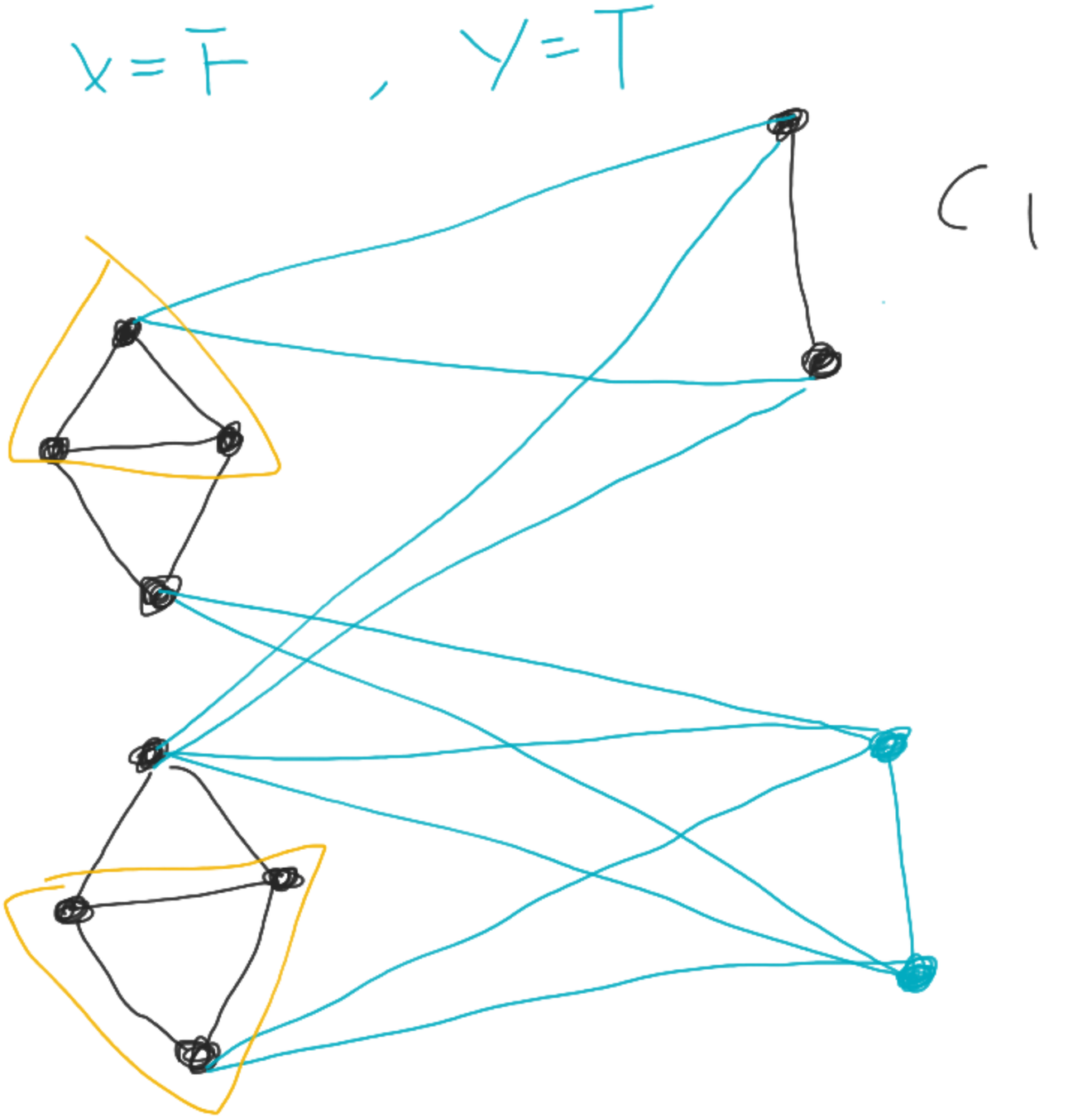


(x or y)

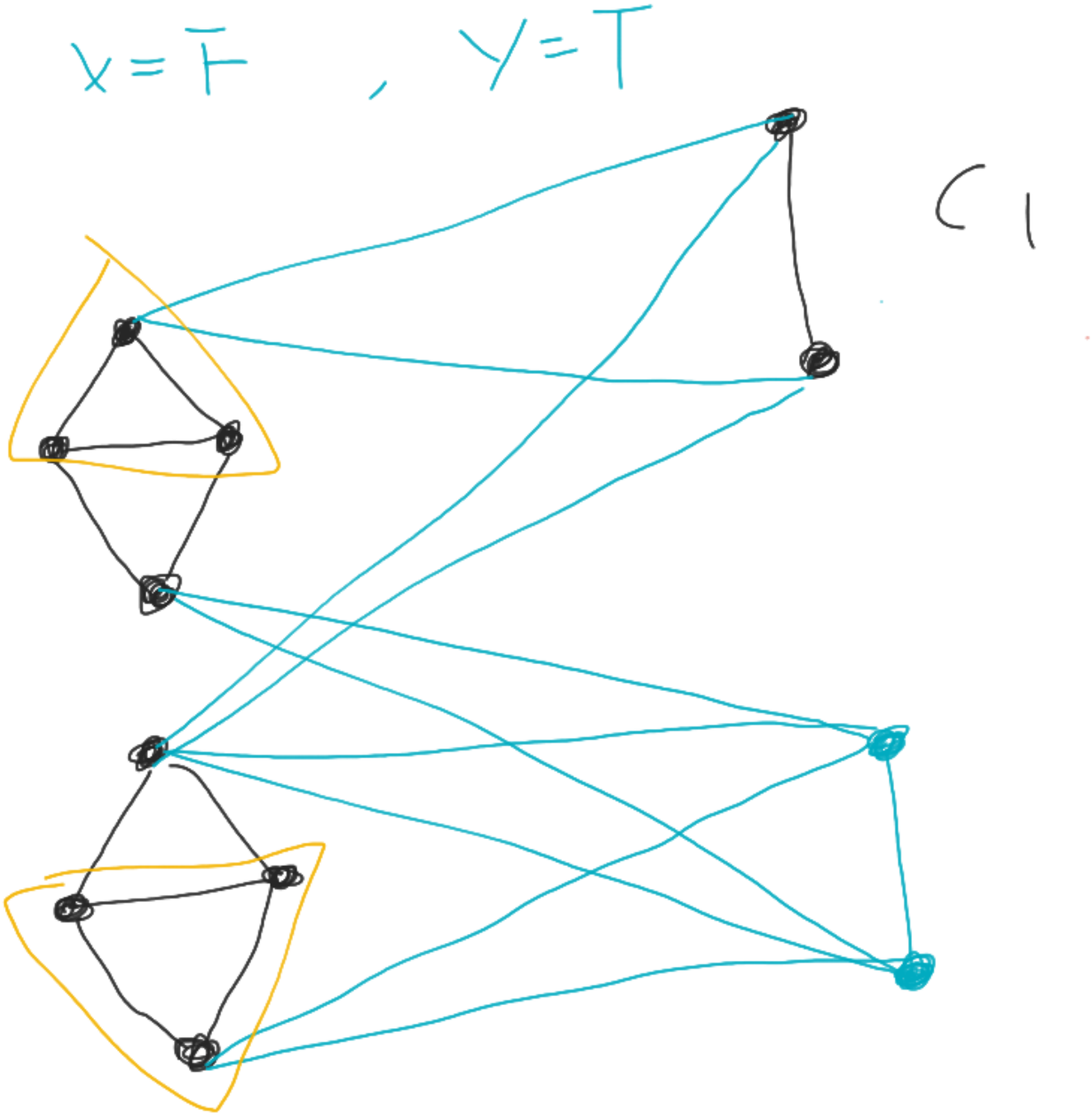
x = T , y = T



(x or y)



(x or y)



6.
Given oracle for HALF-CLIQUE or IS, and G
Return the clique/independent set that proves G in HALF-Clique of IS

You can simply return the nodes that belong to this clique/IS

P on input <G>
  if ORACLE(G) returns false, return null
  let G' be G with modifications
  use the ORACLE to determine whether the graph contains a HALF-CLIQUE or IS.

  for each node in G remove and see if Oracle still passes

Clique Solver given Oracle <G = (V, E)>
  for node n in G:
    if Oracle(V - { n }) let V = V - {n}
  return V

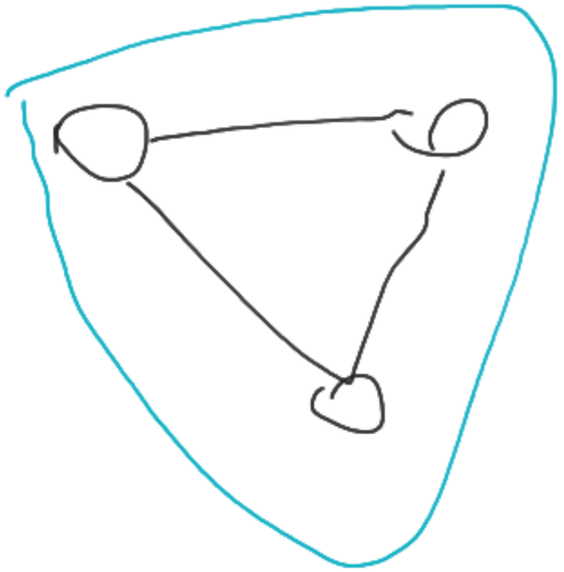
3b

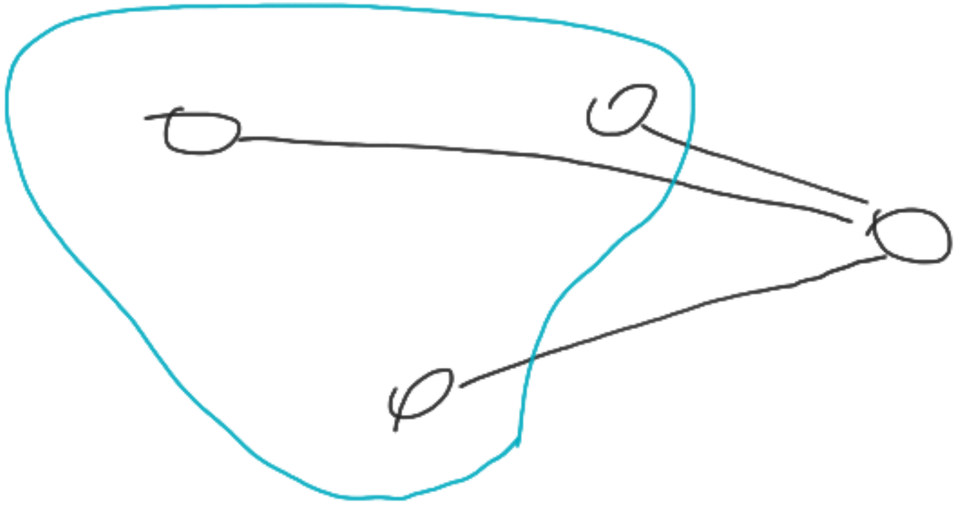
Show many Many-Clique is co-NP-HARD (show Co-Many-Clique is NP Hard)

Clique = NP hard, Co-Many-Clique. Given an instance of Clique, output a graph G such that exists size r subgraph of G does not contain a k-clique

Co-Many-Clique { <G, r, k> | Exists size r subgraph of G does not contain a k-clique }

Given <G, k> , if the input contains a k- clique, we want our output graph to have a size r subgraph that does not contain a k-clique





if such a k clique exists in the original graph, then a size k subset, exists in the complement where no nodes are connected, if no nodes are connected, then there is no 2-clique

Q5
NAE-3SAT

P on input <phi>:
  for each variable x
    run oracle on phi and (x or x or x_aux) and (x_aux, ...)
    if oracle returns true, variable needs to be true
    else variable needs to be false

If an input graph lacks a k clique, then there is no K sized subgraph in the complement such that no edges are in between the nodes.