



Adaptive Algorithms for Complex Group Testing



Hung-Lin Fu 傅 恆 霖

Department of Applied Mathematics

National Chiao Tung University, Hsin Chu, Taiwan

(Work jointly with Huilan Chang and Chih-Huai Shih)



Group Testing

- Consider a set N of n items consisting of at most d *positive* (used to be called defective) items with the others being *negative* (used to be called good) items.
- A group test, sometimes called a *pool*, can be applied to an arbitrary set S of items with two possible outcomes; *negative*: all items in S are negative; *positive*: at least one positive item in S , not knowing which or how many.



Algorithms



- An adaptive (*sequential*) algorithm conducts the tests *one by one* and the outcomes of all previous tests can be used to set up the later test. A *non-adaptive* algorithm specifies a set of tests in advance so that they can be conducted *simultaneously*, thus forbidding using the information of previous tests.



Complex Group Testing



- If the positives are subsets of items instead of just individual items, then we have the complex group testing.
- The subsets are called *complexes*.
- The problems of complex group testing can be formulated as *hidden graphs problems*.





Example of Complex Group Testing

- In screening clone library the goal is to determine which clones in the clone library hybridize with a given probe in an efficient fashion. A clone is said to be **positive** if it hybridize with the **probe(探針)**, and **negative** otherwise.



Goal



- Our goal is to provide an experimental protocol that identifies all pairs (*complex*) of adjacent primers with as few PCRs (queries) (or multiplex PCRs respectively) as possible.



Mathematical Models

Hidden Hypergraphs (Reconstructed)

Where?



Idea



- Edge-detecting queries: $Q_G(S)$
- $Q_G(S) = 1$ if the induced subgraph of the vertex subset S of $V(G)$ contains at least one edge of G and $Q_G(S) = 0$ otherwise.
- For convenience, we assume G is defined on $[1, n] = \{1, 2, \dots, n\}$.

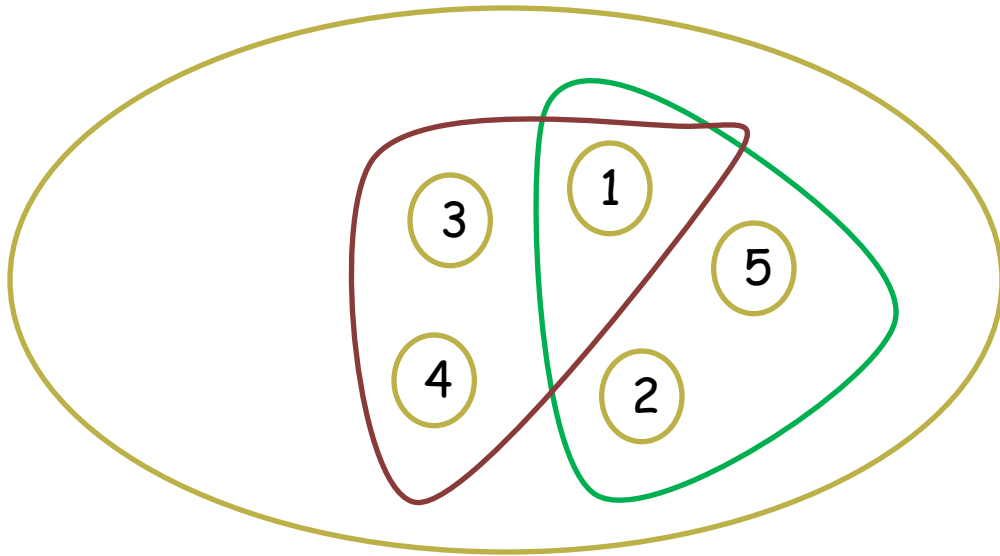


Hidden 3-uniform hypergraphs

- The following example shows an idea of finding a hidden 3-uniform hypergraph.



Find a hyperedge



$\{1,2,3,4,5\}$

$Q(\{1,2,3,4,5\})$

$Q(\{1,2,3\})$

$Q(\{1,2,3,4\})$

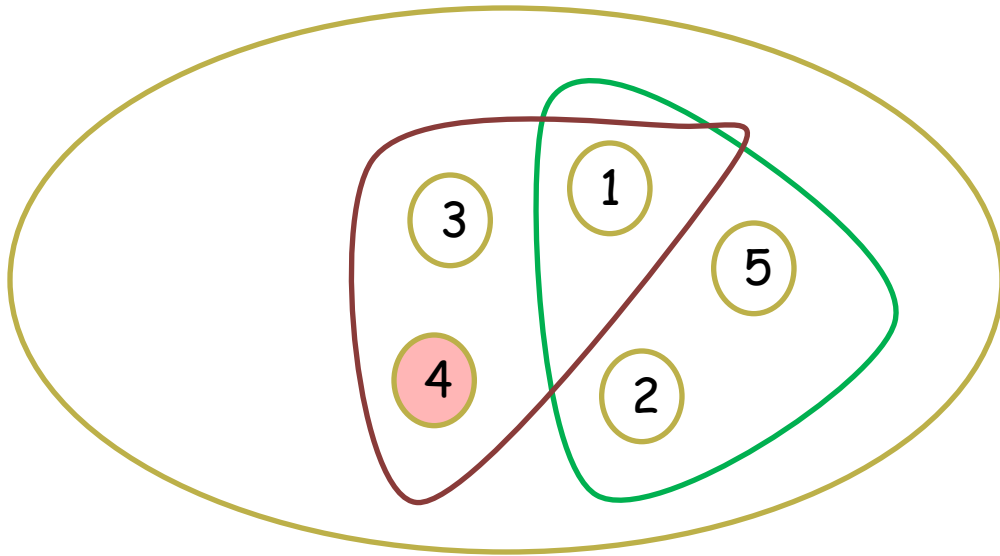
Yes

No

Yes



Find a hyperedge



Algorithm

$\{1,2,3,4,5\}$

$Q(\{1,2,3,4,5\})$

Yes

$Q(\{1,2,3\})$

No

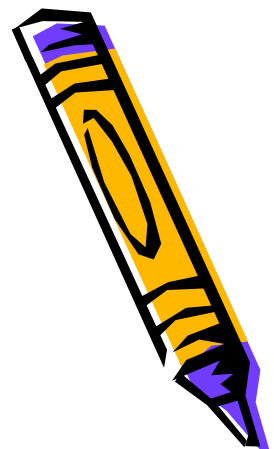
$Q(\{1,2,3,4\})$

Yes

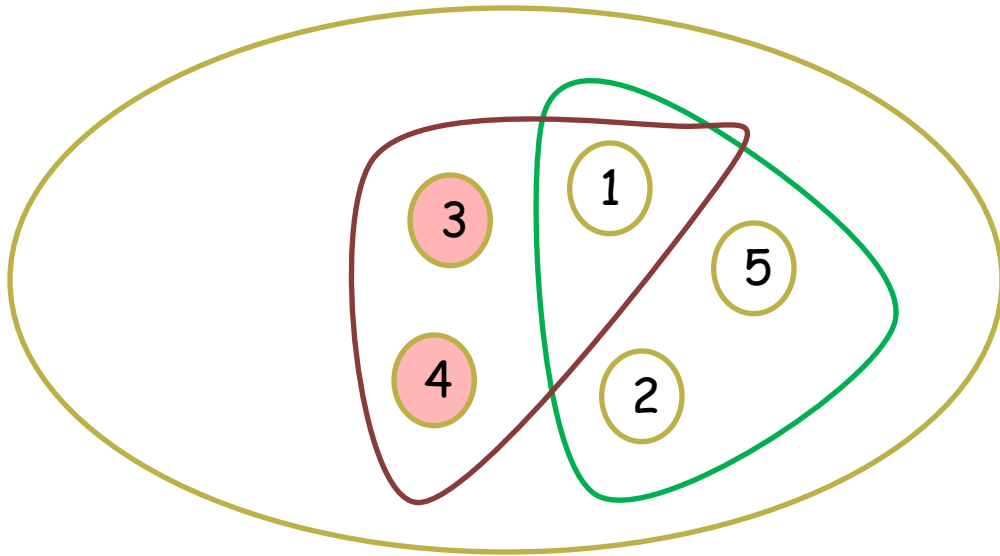
$\{4,1,2,3\}$

$Q(\{4,1,2\})$

No



Find a hyperedge



Algorithm

$\{1,2,3,4,5\}$

$Q(\{1,2,3,4,5\})$

Yes

$Q(\{1,2,3\})$

No

$Q(\{1,2,3,4\})$

Yes

$\{4,1,2,3\}$

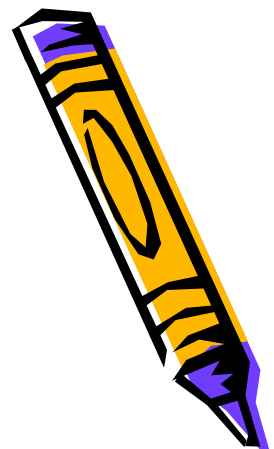
$Q(\{4,1,2\})$

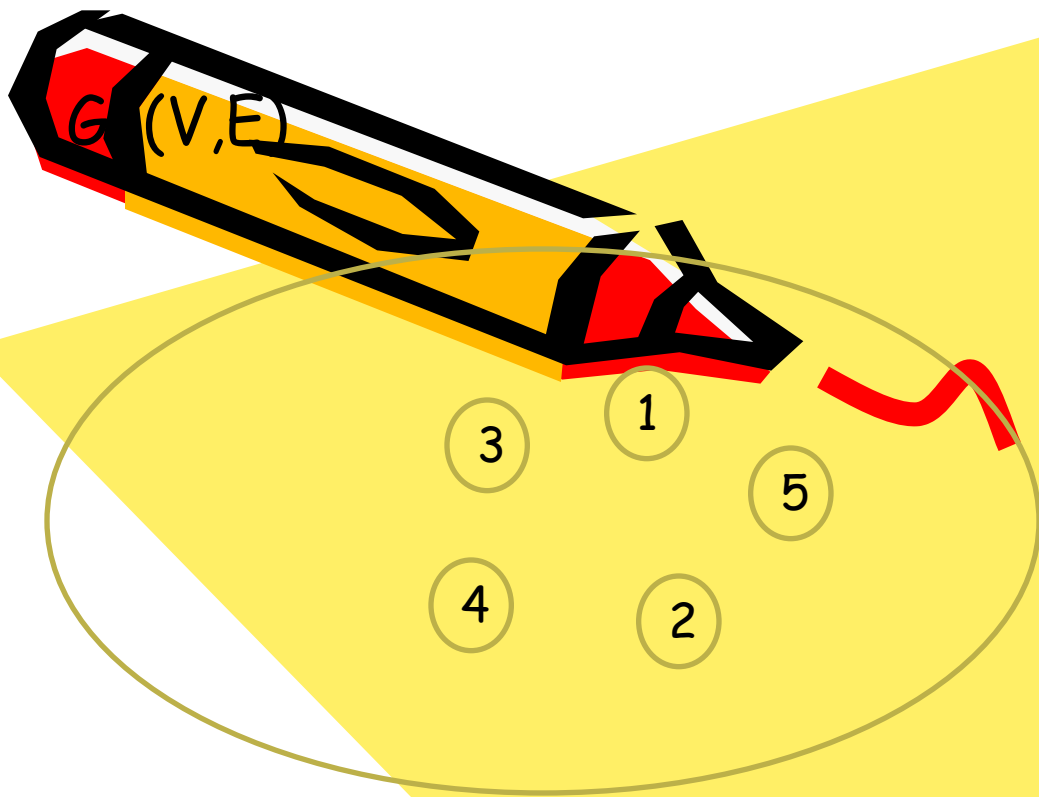
No

$\{4,3,1,2\}$

$Q(\{4,3,1\})$

Yes

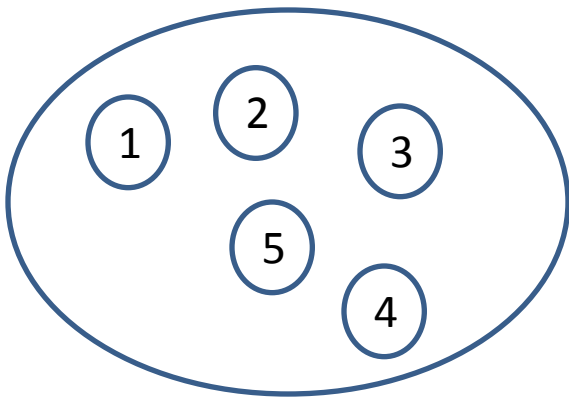




$V = \{1, 2, 3, 4, 5\}$ (more edges)

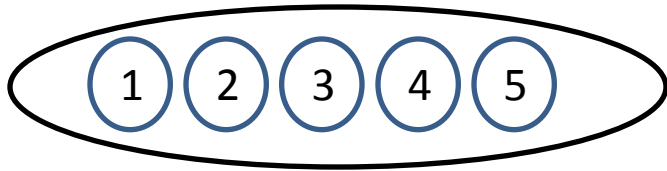
$E = \{\{1, 2, 3\},$
 $\{1, 2, 4\},$
 $\{1, 3, 4\},$
 $\{1, 4, 5\}\}$





$V = \{1, 2, 3, 4, 5\}$

$E = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4, 5\}\}$

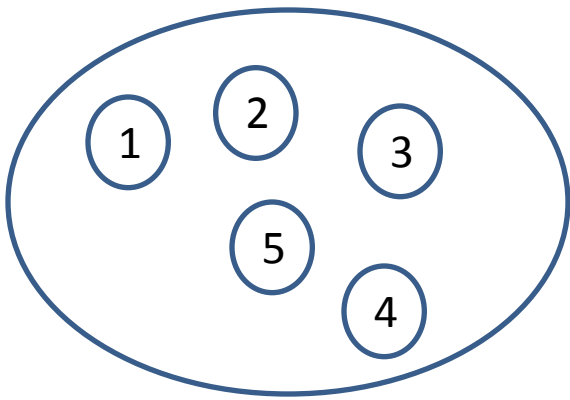


S_1

Initial setting:

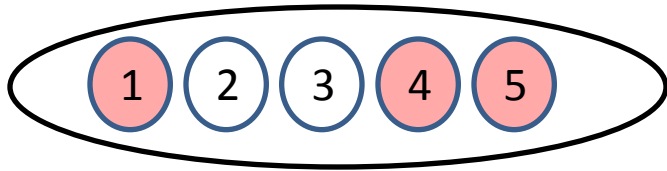
$S_1 = \{1, 2, 3, 4, 5\}$

Find $\{1, 4, 5\}$



$V = \{1, 2, 3, 4, 5\}$

$E = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4, 5\}\}$

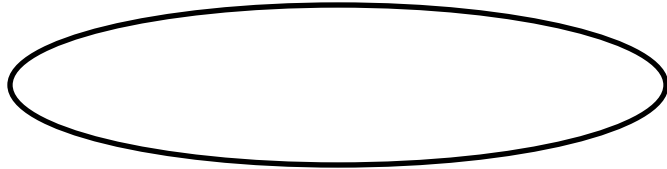
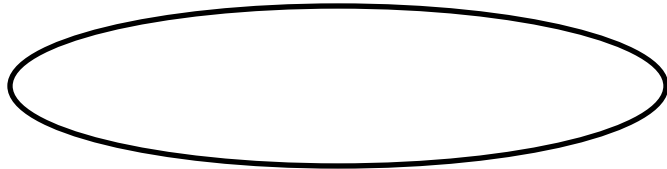


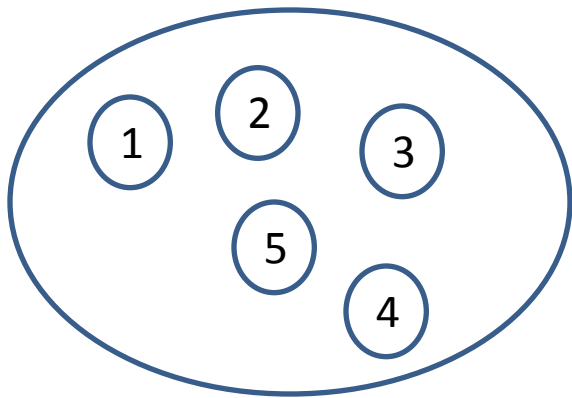
S_0

Initial setting:

$S_0 = \{1, 2, 3, 4, 5\}$

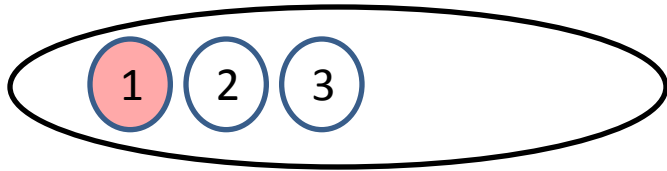
Find $\{1, 4, 5\}$



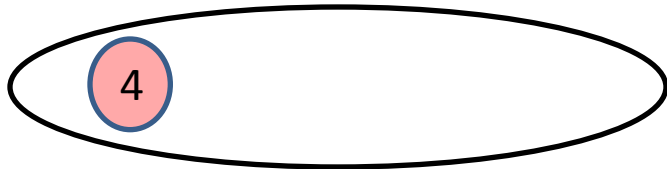


$V = \{1, 2, 3, 4, 5\}$

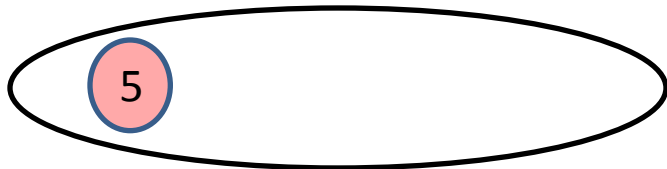
$E = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4, 5\}\}$



S_1



S_2

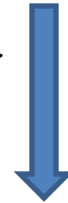


S_3

Initial setting:

$S_1 = \{1, 2, 3, 4, 5\}$

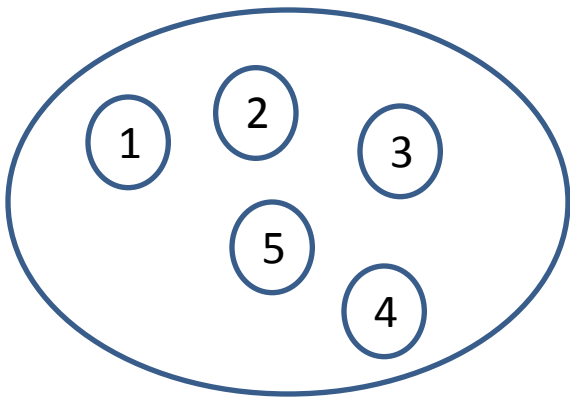
Find $\{1, 4, 5\}$



$S_1 = \{1, 2, 3\}$

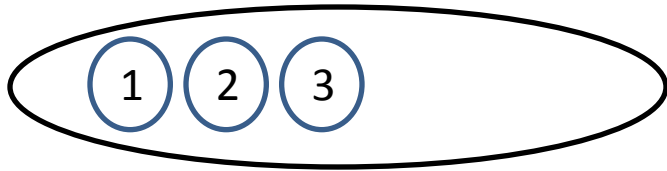
$S_2 = \{4\}$

$S_3 = \{5\}$

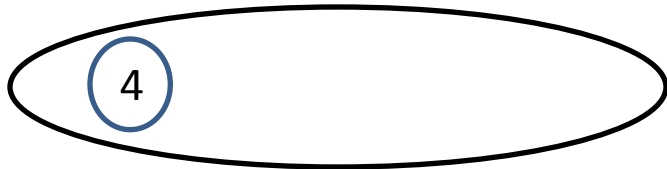


$V = \{1, 2, 3, 4, 5\}$

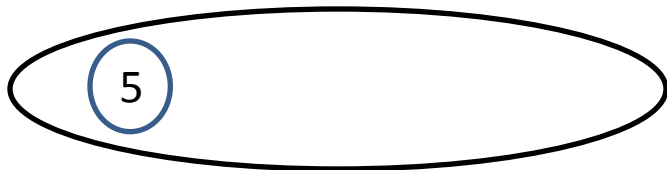
$E = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4, 5\}\}$



S_1



S_2

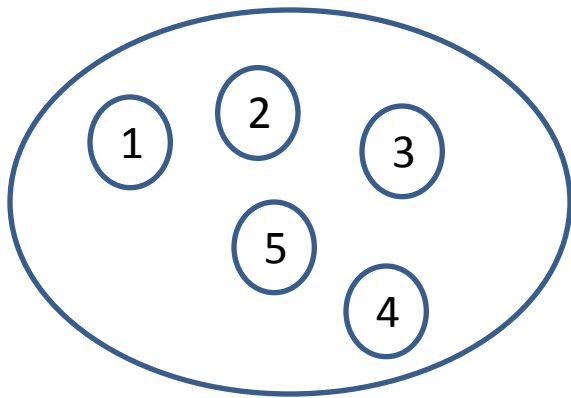


S_3

$S_1 = \{1, 2, 3\}$

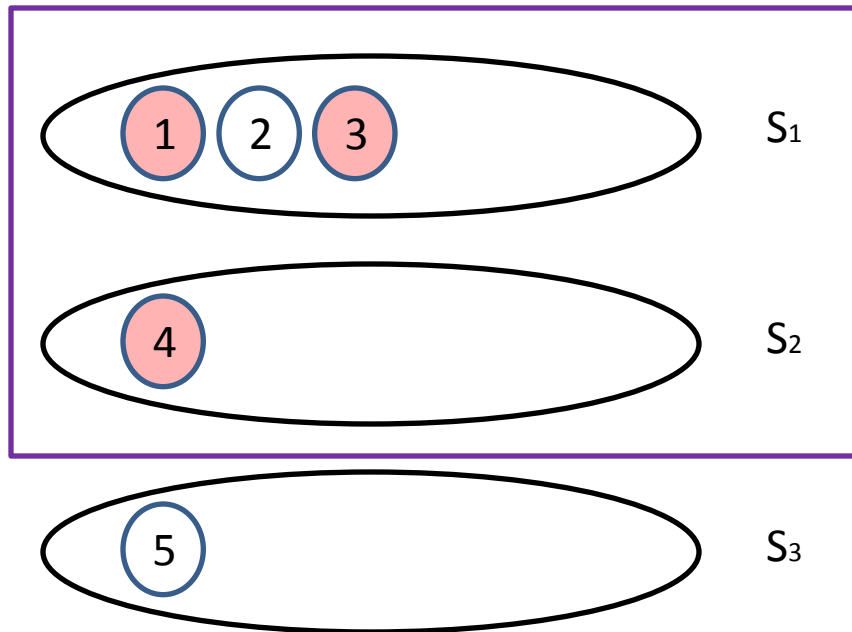
$S_2 = \{4\}$

$S_3 = \{5\}$



$V = \{1, 2, 3, 4, 5\}$

$E = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4, 5\}\}$



$S_1 = \{1, 2, 3\}$

$S_2 = \{4\}$

$S_3 = \{5\}$

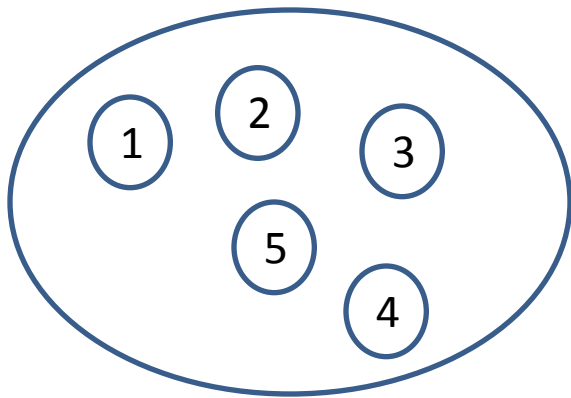
Find $\{1, 3, 4\}$



$S_1 = \{1, 2\}$

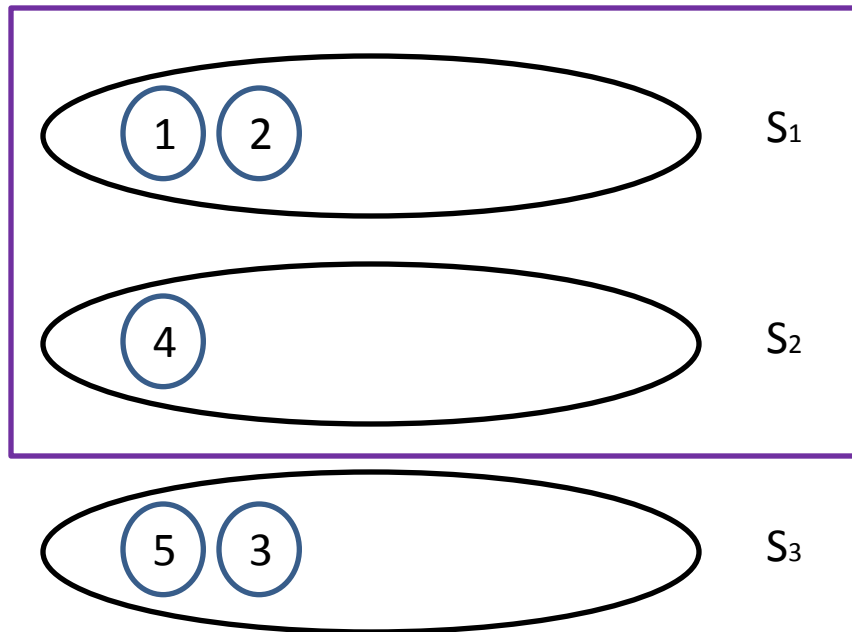
$S_2 = \{4\}$

$S_3 = \{3, 5\}$



$V = \{1, 2, 3, 4, 5\}$

$E = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4, 5\}\}$

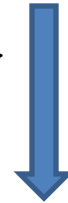


$S_1 = \{1, 2\}$

$S_2 = \{4\}$

$S_3 = \{3, 5\}$

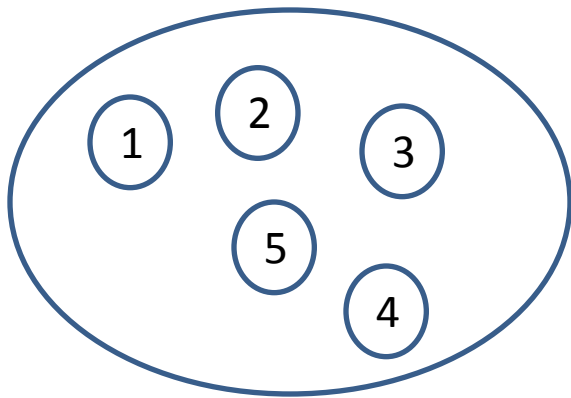
Find $\{1, 2, 4\}$



$S_1 = \{1\}$

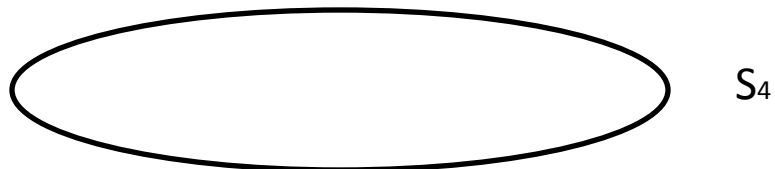
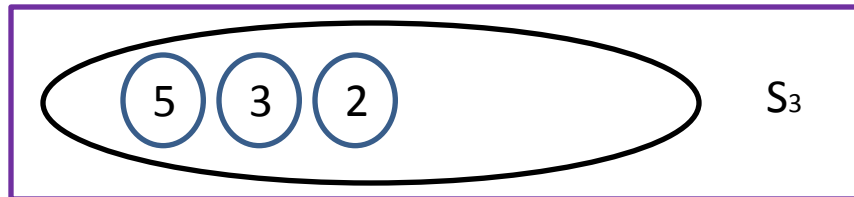
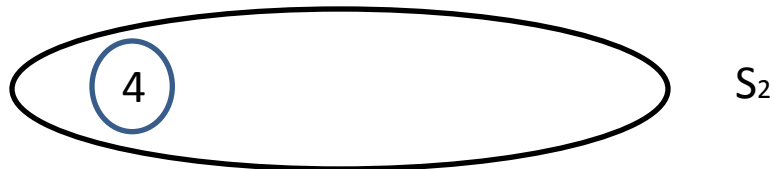
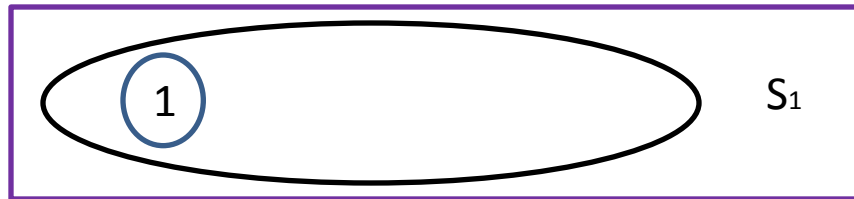
$S_2 = \{4\}$

$S_3 = \{2, 3, 5\}$



$V = \{1, 2, 3, 4, 5\}$

$E = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4, 5\}\}$

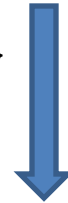


$S_1 = \{1\}$

$S_2 = \{4\}$

$S_3 = \{2, 3, 5\}$

Find $\{1, 2, 3\}$

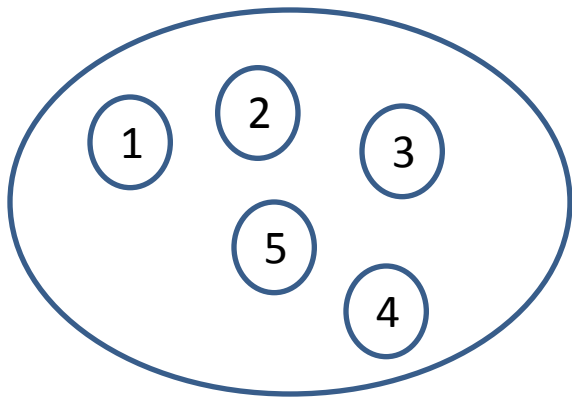


$S_1 = \{1\}$

$S_2 = \{4\}$

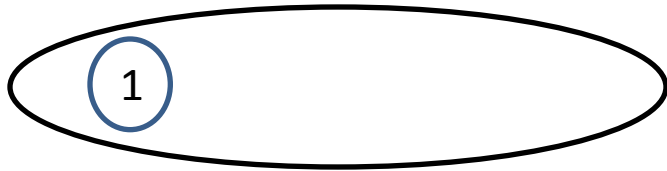
$S_3 = \{2, 5\}$

$S_4 = \{3\}$

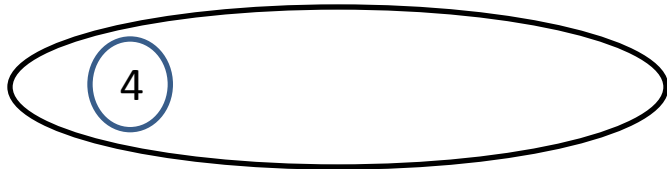


$V = \{1, 2, 3, 4, 5\}$

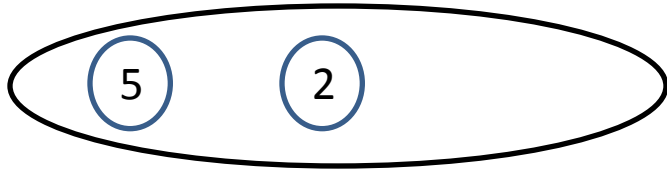
$E = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4, 5\}\}$



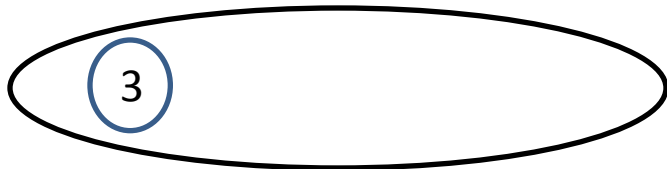
S_1



S_2



S_3



S_4

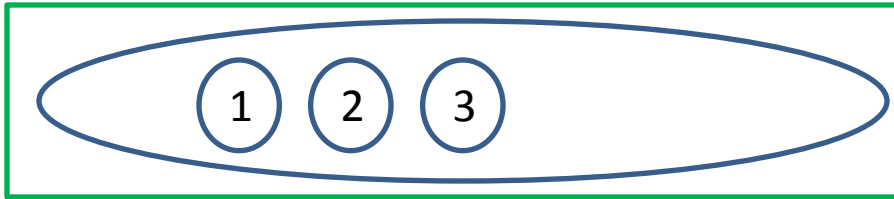
$S_1 = \{1\}$

$S_2 = \{4\}$

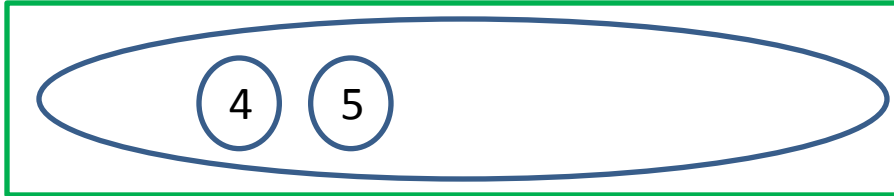
$S_3 = \{2, 5\}$

$S_4 = \{3\}$

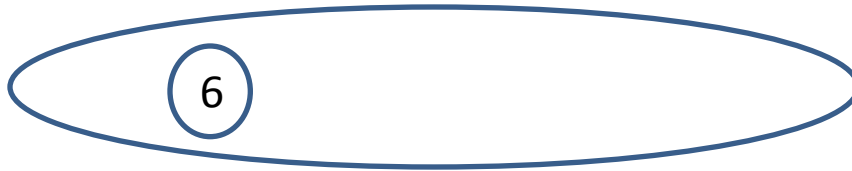




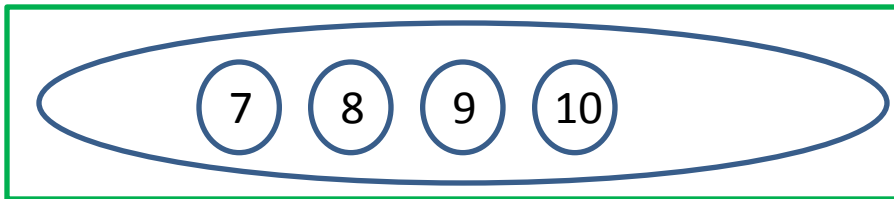
S_1



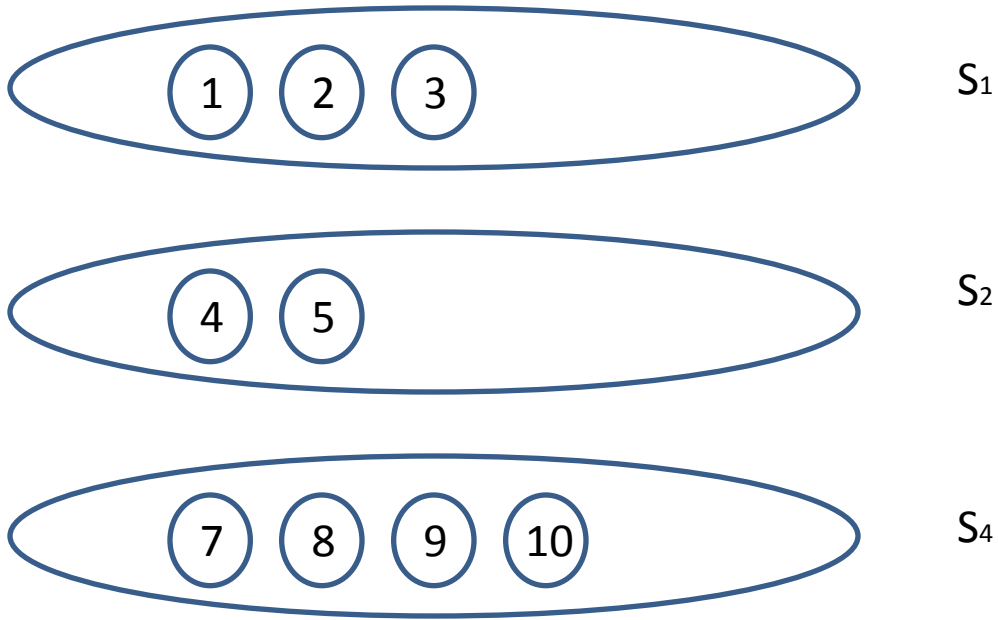
S_2



S_3

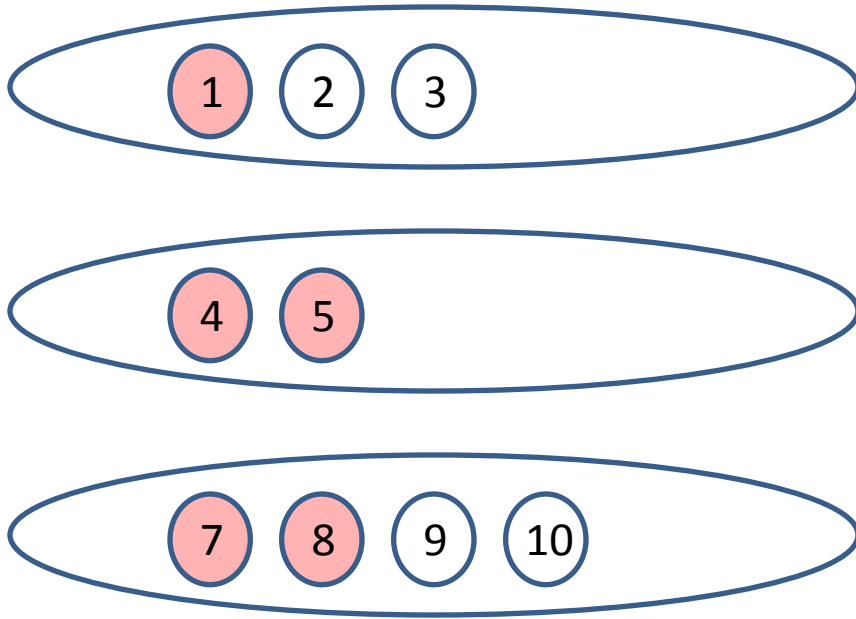


S_4



$$E(G[S_1 \cup S_2 \cup S_4]) = \{\boxed{\{1,4,7\}}, \boxed{\{1,5,8\}}, \{1,4,9\}, \{1,5,7\}\}$$

hidden



S_1

$$Q(S_1 \cup S_2 \cup S_4 - \{1\})$$

NO

S_2

$$Q(S_1 \cup S_2 \cup S_4 - \{4, 5\})$$

NO

S_4

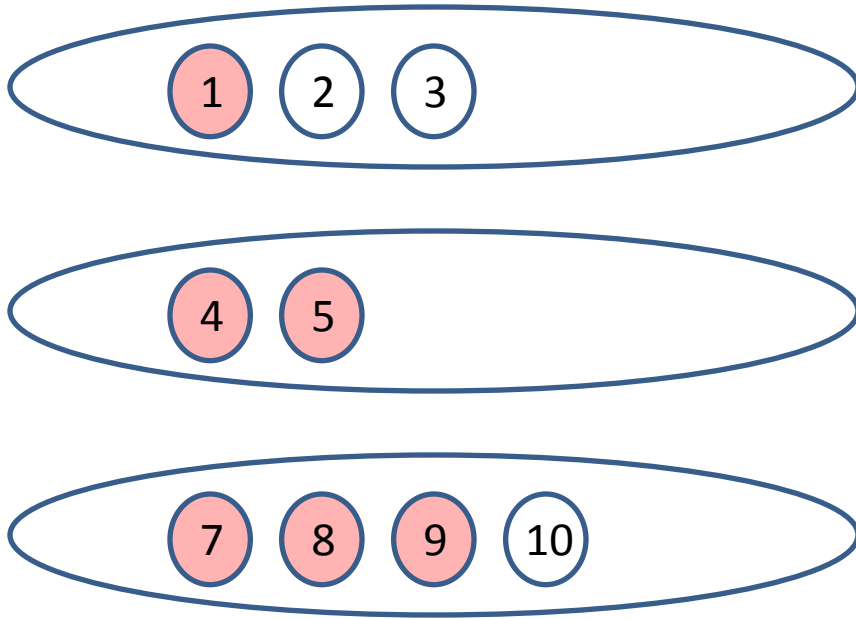
$$Q(S_1 \cup S_2 \cup S_4 - \{7, 8\})$$

Yes

Find $\{1, 4, 9\}$

$$E(G[S_1 \cup S_2 \cup S_4]) = \{\{1, 4, 7\}, \{1, 5, 8\}, \{1, 4, 9\}, \{1, 5, 7\}\}$$

hidden



S_1

$$Q(S_1 \cup S_2 \cup S_4 - \{1\})$$

NO

S_2

$$Q(S_1 \cup S_2 \cup S_4 - \{4,5\})$$

NO

S_4

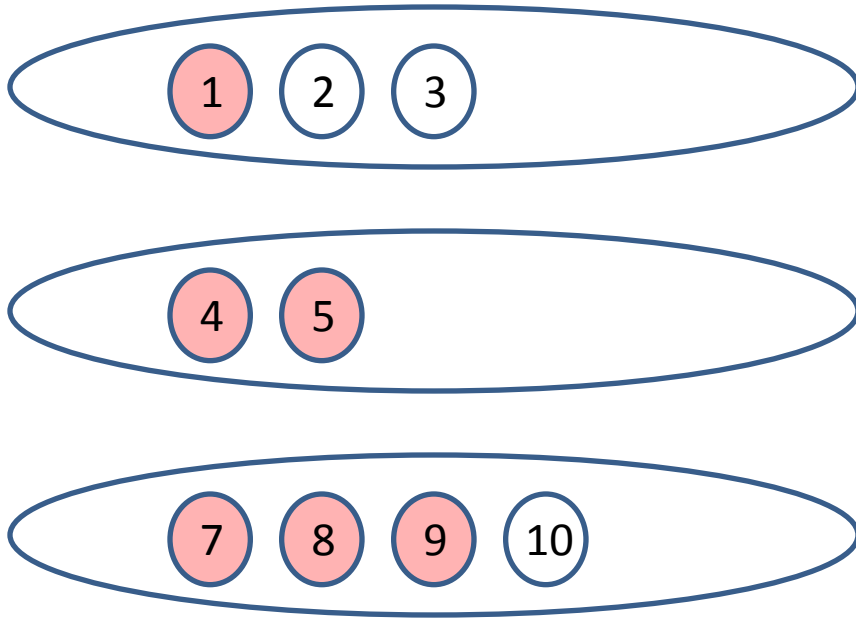
$$Q(S_1 \cup S_2 \cup S_4 - \{7,8,9\})$$

NO

$$E(G[S_1 \cup S_2 \cup S_4]) = \{\{1,4,7\}, \{1,5,8\}, \{1,4,9\}, \{1,5,7\}\}$$



hidden



S_1

$Q(\{1\} \cup \{4\} \cup \{7,9\})$

NO

S_2

$Q(\{1\} \cup \{5\} \cup \{7,9\})$

Yes

S_4

$Q(\{1\} \cup \{5\} \cup \{7\})$ **Yes**

DONE!

$$E(G[S_1 \cup S_2 \cup S_4]) = \{\{1,4,7\}, \{1,5,8\}, \{1,4,9\}, \{1,5,7\}\}$$



hidden

Conclusion



- We can find a hidden r -uniform hypergraph G with $m \cdot r \cdot \log_2 n + (2^{r+2})^{1/2} r^r m^{r/2}$ edge-detecting queries where m is the number of edges in G .
- Note that if $r = 2$, then we can find a hidden graph with at most $m \cdot 2 \cdot \log_2 n + 9m$ queries. (with Huilan Chang and Chih-Huai Shih, Optimization Letters 2014)



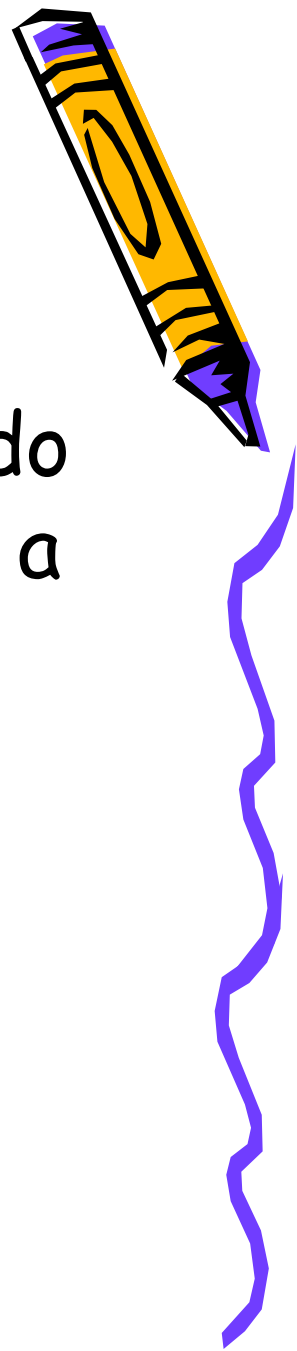
Problems



- We believe that **the best** answer should be around $(m \log_2 n)$ queries, but not able to prove this fact.
- We have no **good** idea about **non-adaptive algorithm** for optimal result at this moment.



More Problems



- If there is an *inhibitor*, then how do we use *adaptive algorithm* to learn a hidden graph?
- What if *errors* occurred when *adaptive algorithm* is utilized?



Some references



- Identification and classification problem on pooling designs for *inhibitor* models (with Huilan Chang and Hong-Bin Chen), **J. Computational Biology**, 17(2010), No. 7, 927-941.
- Reconstruction of *hidden graphs and threshold* group testing (with Huilan Chang, Hong-Bin Chen and Chi-Huai Shih), **J. Combin. Optimization**, 22(2011), no. 2, 270 - 281.
- Group testing with multiple *mutually-obscuring positives* (with Hong-Bin Chen), **Lecture Notes Computer Science**, 7777(2013), 557 - 568.
- *Threshold* group testing on *inhibitor* model (Huilan Chang and Chie-Huai Shih), **J. Computational Biology**, Vol. 20, No. 6, 2013, 1 - 7.

