# *Adaptive Algorithms in Group Testing*

Hung-Lin Fu 傅 恒 霖

交通大學應用數學系

Hsin Chu, Taiwan

# Research Facts

- The idea of group testing was proposed by a team member (Rosenblatt) of R. Dorfman and Dorfman went ahead to write a journal paper to emphasize the importance of this idea.
- The detection of defective members of large populations, Ann. Math. Statist. 14 (1943), 436 – 440.
- Upon the report, this idea has never been used then. ***People are not confident about it!***

# Information to Know

- Hsin Chu played the most important role in the production of Christmas light bulbs. (Before the construction of Science Based park).
- People lived along 光復路 make the key contribution!
- They knew how to apply ***Group Testing*** already.
- Currently, more than 80% such light bulbs are manufactured by a **Taiwanese company** located in mainland China.

# 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 or pure) 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 one or how many.

# Adaptive (Sequential) and Non-adaptive

- ***Adaptive (or sequential) algorithm***: You ask the second question (query) after knowing the answer of the first one and continue …. That is, the previous knowledge will be used later.
- ***Non-adaptive (or parallel) algorithm***: You ask all the questions (queries) *simultaneously, and then use all the outcomes to make a conclusion.*

# Algorithms

- Adaptive algorithm (***Smart idea is needed!***)
- Non-adaptive algorithm (***Construction of Beautiful Arrays!***)
- ***k-stage algorithm***

The most popular one is a 2-stage algorithm in which we use an algorithm (non-adaptive) in the first stage and then in the second stage we test the left "suspected" items (from the first stage) one by one.

# Fundamental Problem

- Among all possible algorithms let M(d, n) denote the minimum number of tests we need to use in the worst case, for example, M(1, 64) = 6 (?).
- ***Problem: Determine M(d, n).***
- ***Fact***: $d \cdot \{\log_2 n\} \geq M(d, n) \geq \{\log_2 C(n,d)\}$.
  (C(n,d) denotes n-choose-d.)
- The inequality of the right hand side is known as the information lower bound.

# Development of GT (1)

- - 1999 Complex model: Molecular biology (Torney)
- - 1998 Mutually obscuring defetives (Damaschke)
- - 1997 Inhibitor model (Farach *et al*.)
- - 1993 Combinatorial group testing (D. Z. Du and Frank K. Hwang）
- - 1984 Multiple-access communication (Berger et al.)
- - 1943 Blood testing (Dorfman & Rosenblatt)

# Development of GT (2)

- 2006 – Present: ***Many others!***
- 2006 Pooling designs and nonadaptive group testing (D. Z. Du and Frank K. Hwang).
- ***Graph reconstruction (contig sequencing)***
- Non-unique probe selection problem
- - 2006 Inhibitor complex model (Chang *et al*.)
- - 2001 Drug discovery (Xie *et al.*)
- - 2000 Combinatorial group testing, 2ed (D. Z. Du and Frank K. Hwang)

# Various Models

- *Errors* occurred (*in experiments*).
- There are *inhibitors (cancel the positive effects)*.
- There is a *threshold (enough positives)*.
- Defective items are sets: *complex* model.
- *Competitive* model: the number of defectives is unknown.
- The defectives are *mutually obscuring*.

# Various Approaches

- There is a limitation of pool sizes.
- Testing items are arranged along a path and the set of all items are arranged as the vertices of a graph.
- To determine the positivity of tests, there is a ratio for reference, say 1/12. (Density model)
- Probabilistic model: each item has a probability to be positive.

# Further Remarks

- We can approach this study via different topics such as *combinatorial design, algebraic combinatorics, coding theory and graph theory*, and of course ***algorithms***.
- Group testing does play an important role in applications such as *computational molecular biology, network security, image compression, data patterns, fingerprinting in multimedia*, …, etc.

# Adaptive Algorithms

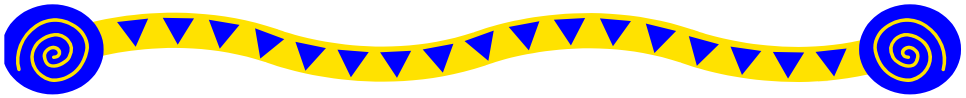
- $M(1, n) = \{\log_2 n\}$. (Ceiling)
- Determining M(2, n) in general is still open.
- Generalized binary splitting algorithm (Frank K. Hwang) showed that M(d, n) is either *$\{\log_2 C(n, d)\}$ or $\{\log_2 C(n, d)\} + 1$ . . . Or $\log_2 C(n, d)\} + d - 1$ . (1972, J. ASA)*
- The idea (for general d) comes from selecting ***a subset*** of suitable size $2^{\alpha}$ for the first test.

  **($\alpha = \{\log_2 m/d\}$, $m = n - d + 1$.)**

# Best Known Results

- By using the generalized binary splitting algorithm of Frank K. Hwang, we are able to show that $|M(d, n) - \{\log_2 C(n, d)\}| \leq d - 1$.

Our goal is to tighten this gap.



Closer!

# Quaternary Splitting Algorithm

- For d = 2, our method does not have a better result since the gap “1” is the best so far.
- May be some of you can settle this case and find the exact solution for M(2, n). (Gambate!)
- We believe the answer should be a characterization of this value, that is, to show: *$M(2, n) = \{\log_2 C(n, 2)\}$ for certain n and $\{\log_2 C(n, 2)\} + 1$ otherwise.*

# The idea of our algorithm "L"

- Step 1. For each n, determine whether

$$14 \cdot 2^{k-1} < n \leq 10 \cdot 2^k \text{ or}$$

$$10 \cdot 2^k < n \leq 14 \cdot 2^k.$$

- Step 2. (Consider $n = 10 \cdot 2^k$ and $n = 14 \cdot 2^k$)

Partition the whole set S of size n into four subsets $S_0$, $S_1$, $S_2$ and $S_3$ with sizes ratio 1:2:2:5 and 1:3:3:7 respectively. Let the first test $T_1$ be on the set $S_0 \cup S_1$.

# Continued …

If $T_1$ is negative (N), then set n as $|S_2 \cup S_3|$ and go back to step 1. On the other hand, go to Step 3.

- Step 3. Let $T_2$ be on the set $S_0 \cup S_2$. If $T_2$ is negative (N), then set n as $|S_1 \cup S_3|$ with $S_1 = S_0' \cup S_1'$ and $S_3 = S_2' \cup S_3'$ . (Keep these sets with the same ratio of sizes as in Step 2.) Return to Step 3 with new partition. If $T_2$ is positive (P), then go to Step 4.

# Continued …

- Step 4. Let $T_3$ be on the set $S_1$. If $T_3$ is N, then find one defective in $S_0$ first and the other one in $S_0 \cup S_2 \cup S_3$. If $T_3$ is P, then find one defective in $S_0 \cup S_2$ and the other in $S_2$. (Using binary splitting algorithm here.)

## Theorem

***$M_L(2, 10 \cdot 2^k) = 2k + 6$ and***

***$M_L(2, 14 \cdot 2^k) = 2k + 7$ for $k \geq 0$.***

# Conclusion

- Corollary

  $M(2, n)$ is at most $2k + 6$ if $14 \cdot 2^{k-1} < n \leq 10 \cdot 2^k$

and $M(2, n)$ is at most $2k + 7$ if $10 \cdot 2^k < n \leq 14 \cdot 2^k$.

- The result obtained here is roughly about the same as known results so far.
- The case $M(2, n)$ is settled if $n$ is of the forms either $10 \cdot 2^k$ or $14 \cdot 2^k$. But, between the gap, some more effort must be made.

# Further remark

- This algorithm also shows that the gap between M(2, n) and $\{\log_2 C(n,2)\}$ is 1.
- We can also show that the gap between M(3, n) and $\{\log_2 C(n,3)\}$ is 1 instead of 2 (which was obtained by using Generalized binary splitting algorithm).
- Hopefully, "d – 2" in general cases of M(d, n).

# d = 3

- Now, we have three intervals to consider:

  $30 \cdot 2^{k-1} < n \leq 19 \cdot 2^k$ or $19 \cdot 2^k < n \leq 24 \cdot 2^k$ or

  $24 \cdot 2^k < n \leq 30 \cdot 2^k$.

- Similar steps except we have ratio of sizes of the four sets: 1:3:3:12, 1:4:4:15, and 1:5:5:19 respectively. ($15 = 30 \cdot 2^{-1}$)
- *Reductive argument is used!*

# Results

***Theorem***

$M_L(3, 19 \cdot 2^k) = 3k + 11,$

$M_L(3, 24 \cdot 2^k) = 3k + 12,$

$M_L(3, 30 \cdot 2^k) = 3k + 13$ for $k \geq 0.$

$M(3, 19 \cdot 2^k) \leq 3k + 11,$

$M(3, 24 \cdot 2^k) \leq 3k + 12,$

$M(3, 30 \cdot 2^k) \leq 3k + 13$ for $k \geq 0.$

# General d

- The d intervals are defined accordingly.
- For d = 4, we have 31, 37, 44 and 52 with ratios of the sizes of 4 sets: 1,4,4,22; 1,5,5,26; 1,6,6,31; 1,7,7,37.
- 1,d,d,(3d-1)d/2 (= r); 1,d+1,d+1,r+d; 1,d+2,d+2,r+2d+1; 1,d+3,d+3,r+3d+3; …;

  1,2d-1,2d-1,2r-2d+1.
- ***To be continued!***

# d = 4

***Theorem***

$M_L(4, 31 \cdot 2^k) = 4k + 17,$

$M_L(4, 37 \cdot 2^k) = 4k + 18,$

$M_L(4, 44 \cdot 2^k) = 4k + 19,$

$M_L(4, 52 \cdot 2^k) = 4k + 20$ *for* $k \geq 0.$

- *For d = 3 and 4, we have*

$|M(d, n) - \{\log_2 C(n, d)\}| \leq d - 2.$

# Complex Model

- An extension of classical group testing problem.
- A set of *complexes*, each of which is a subset of basic items, is given and the property (*positive or negative*) of each complex is waiting to be determined.
- The corresponding group test determines whether a subset contains at least one positive complex (instead of just one positive item).

# Example of Complex Model

- 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.

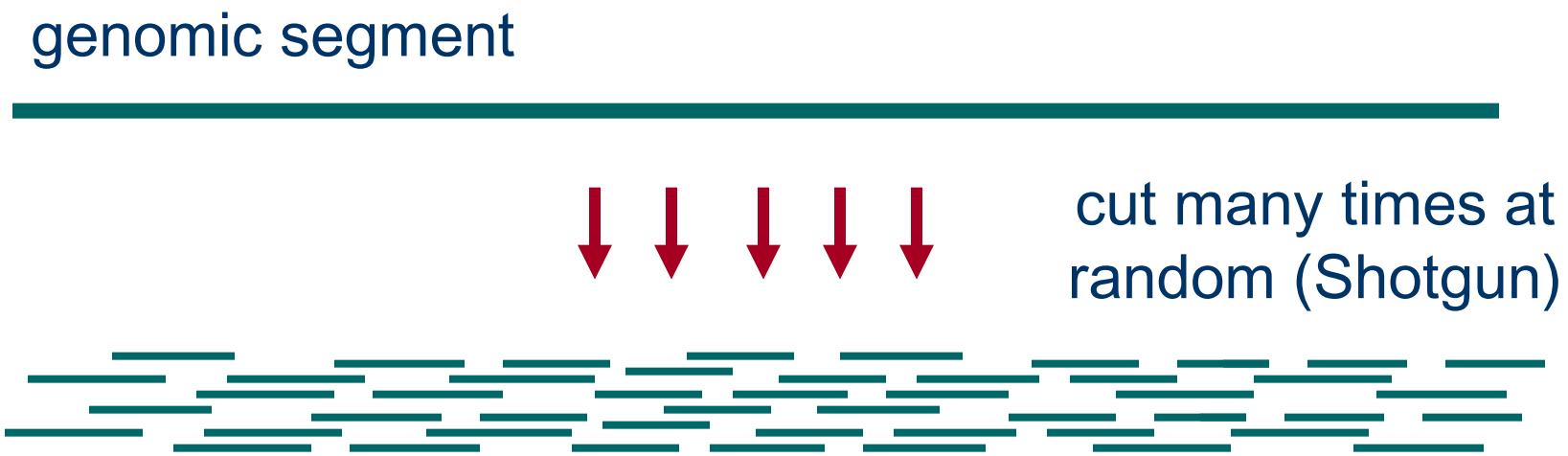
# Whose Idea?

- In 1996, *A. Sorokin et al*. proposed the following idea in *Genome Research*: For physical mapping, some cloned fragments are produced to cover the whole DNA molecule with *some gaps* and then some of them are assembled to form longer continuous fragments (contigs). However, the information about the *mutual placement* of the contigs on the DNA sequence is lost.

# Shotgun Sequencing

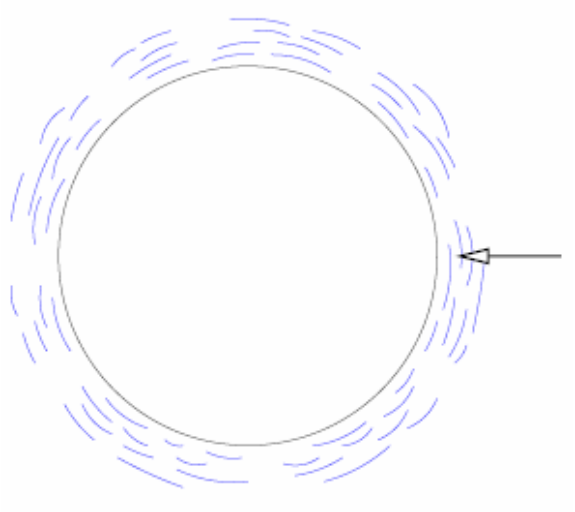
- Shotgun sequencing is a throughput technique resulting in the sequencing of a large number of bacterial genomes, mouse genomes and the celebrated human genomes. In all such projectss, we are left with a collection of contigs that for special reasons cannot be assembled with general assembly algorithms. Continued …

# Shotgun approach



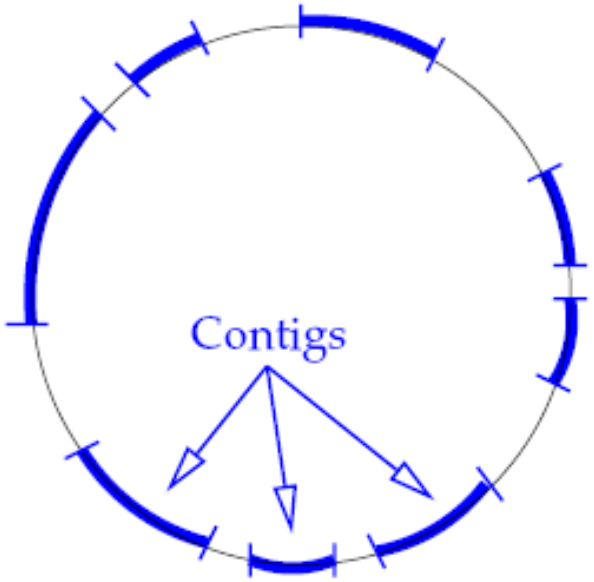
# Whole-genome shotgun sequencing

– Short reads are obtained  and covering the genome with redundancy and possible gaps.

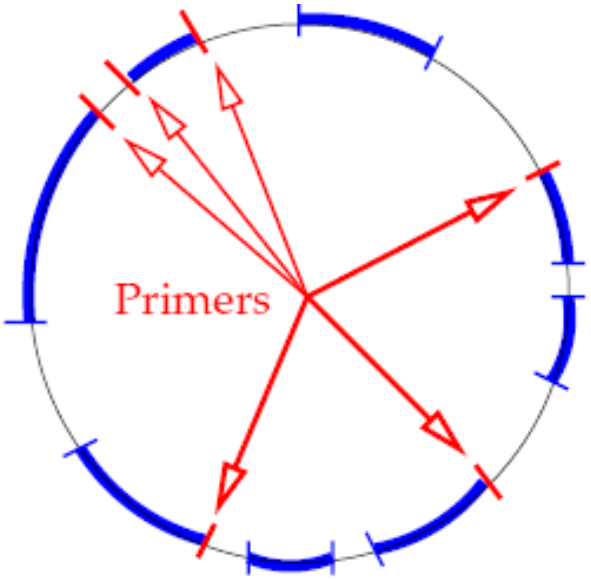


Circular genome

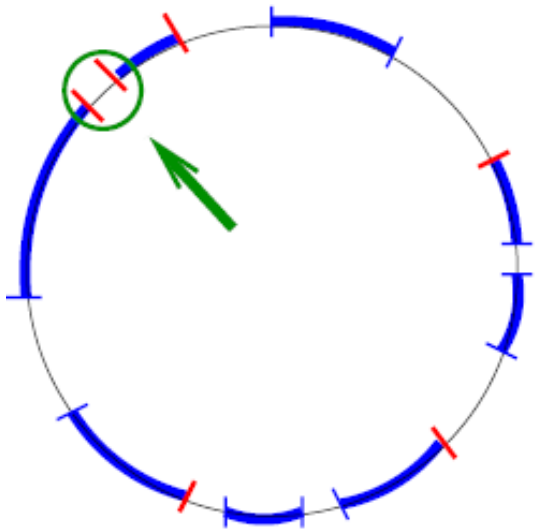
– Reads are assembled into contigs with unknown relative placement.

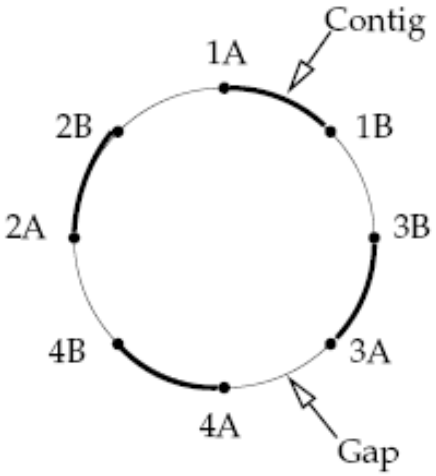


– Primers : (short) fragments of DNA characterizing ends of contigs.

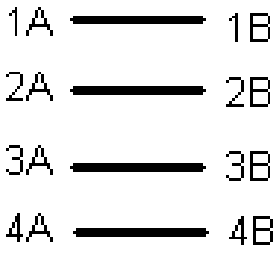


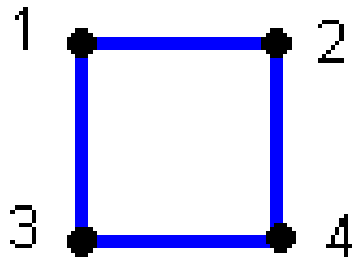
- A PCR (Polymerase Chain Reaction) reaction reveals if two primers are proximate (adjacent to the same gap).
- Multiplex PCR can treat multiple primers simultaneously and outputs if there is a pair of adjacent primers in the input set and even sometimes the number of such pairs.

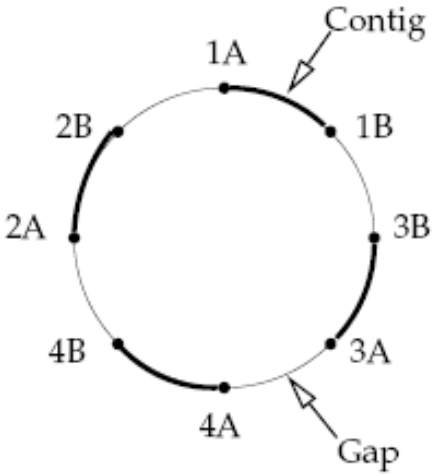


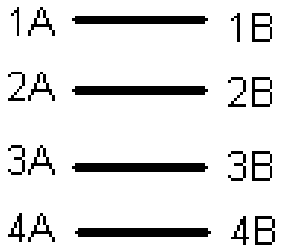



- Two primers of each contig are “mixed together”
  - Find a Hamiltonian cycle by PCRs!

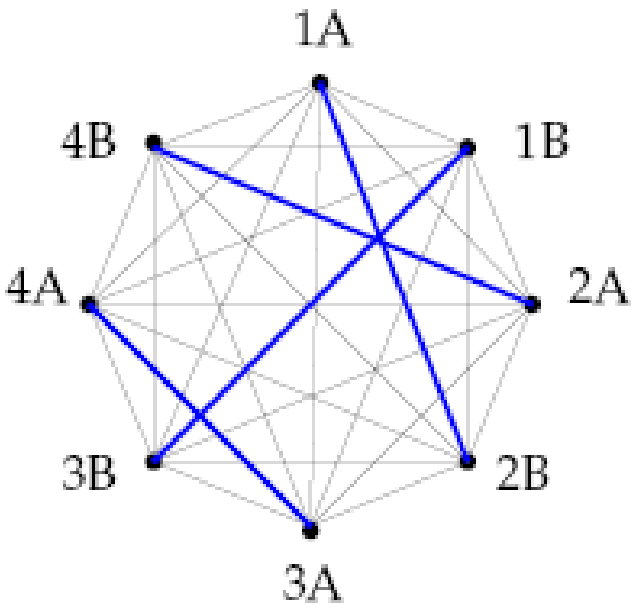





- Primers are treated independently.
  - Find a perfect matching by PCRs.



# Goal

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

# Mathematical Models

*Hidden Graphs or Hypergraphs*

*(Reconstructed)*

- Topology-known graphs, e.g. Hamiltonian cycle, matching, star, clique, bipartite graph, …, etc.
- Graphs of bounded degree
- Hypergraphs
- Graphs of known number of edges

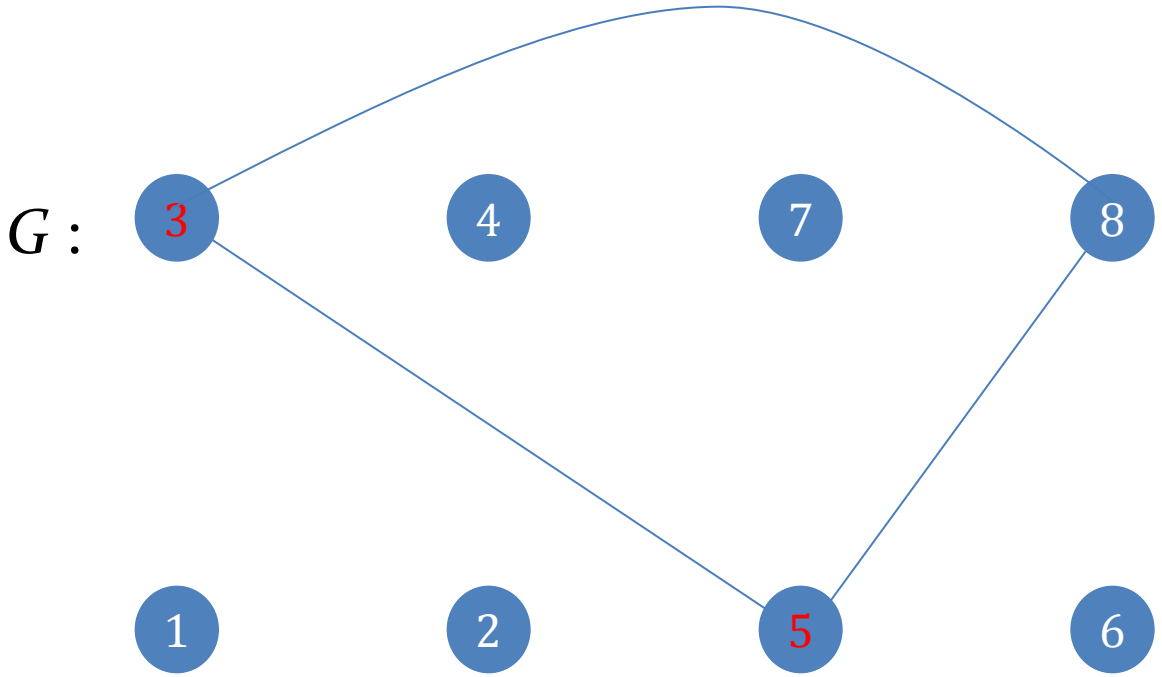
# Goal

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

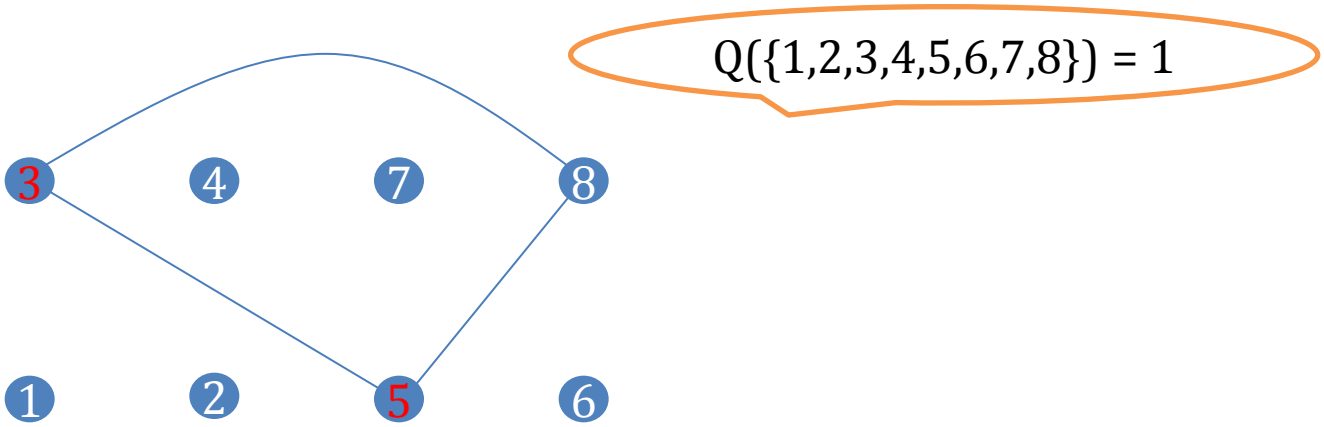
# Graph Learning Problem

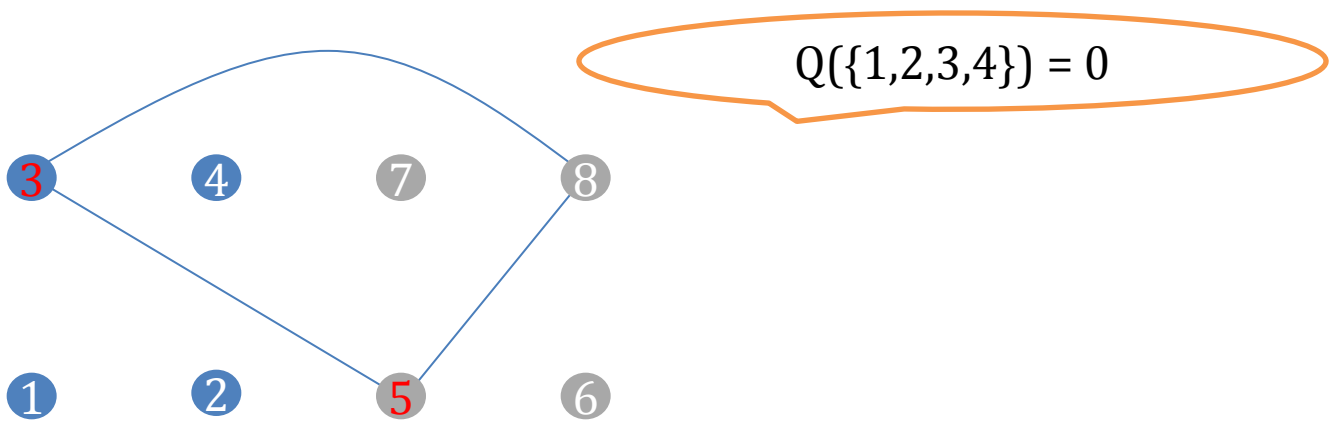
- Let ***H*** be a family of labeled graphs or hypergraphs *(hidden)* defined on the set [n] = {1, 2, …, n}.
- Referring to the information of "*belonging to H*", we wish to identify G by *edge-detecting queries*, each of which tells whether a subset (with some constraints) of [n] induces an edge or a hyper-edge of G.

# *Example (A triangle is hidden)*

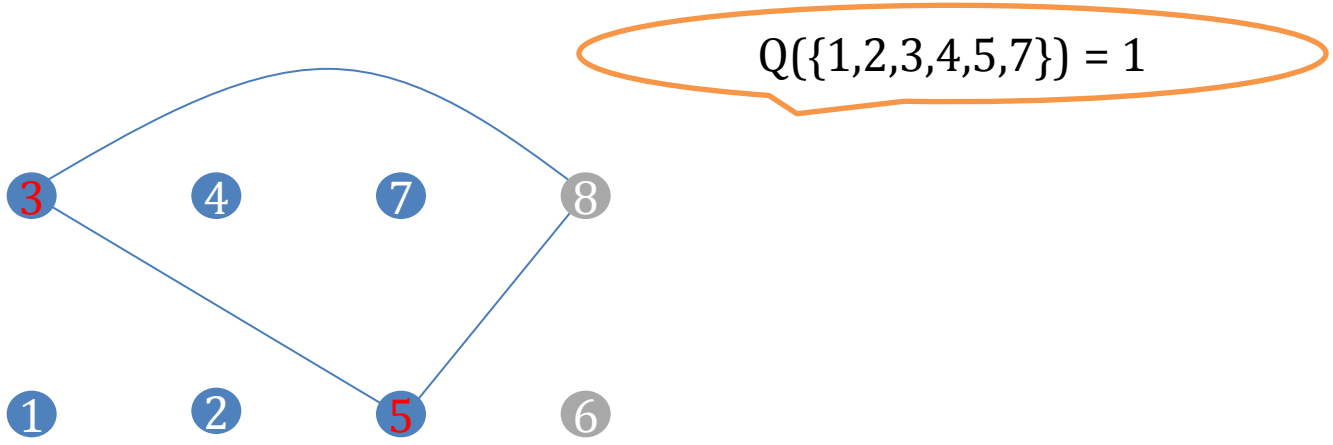


# Find a vertex

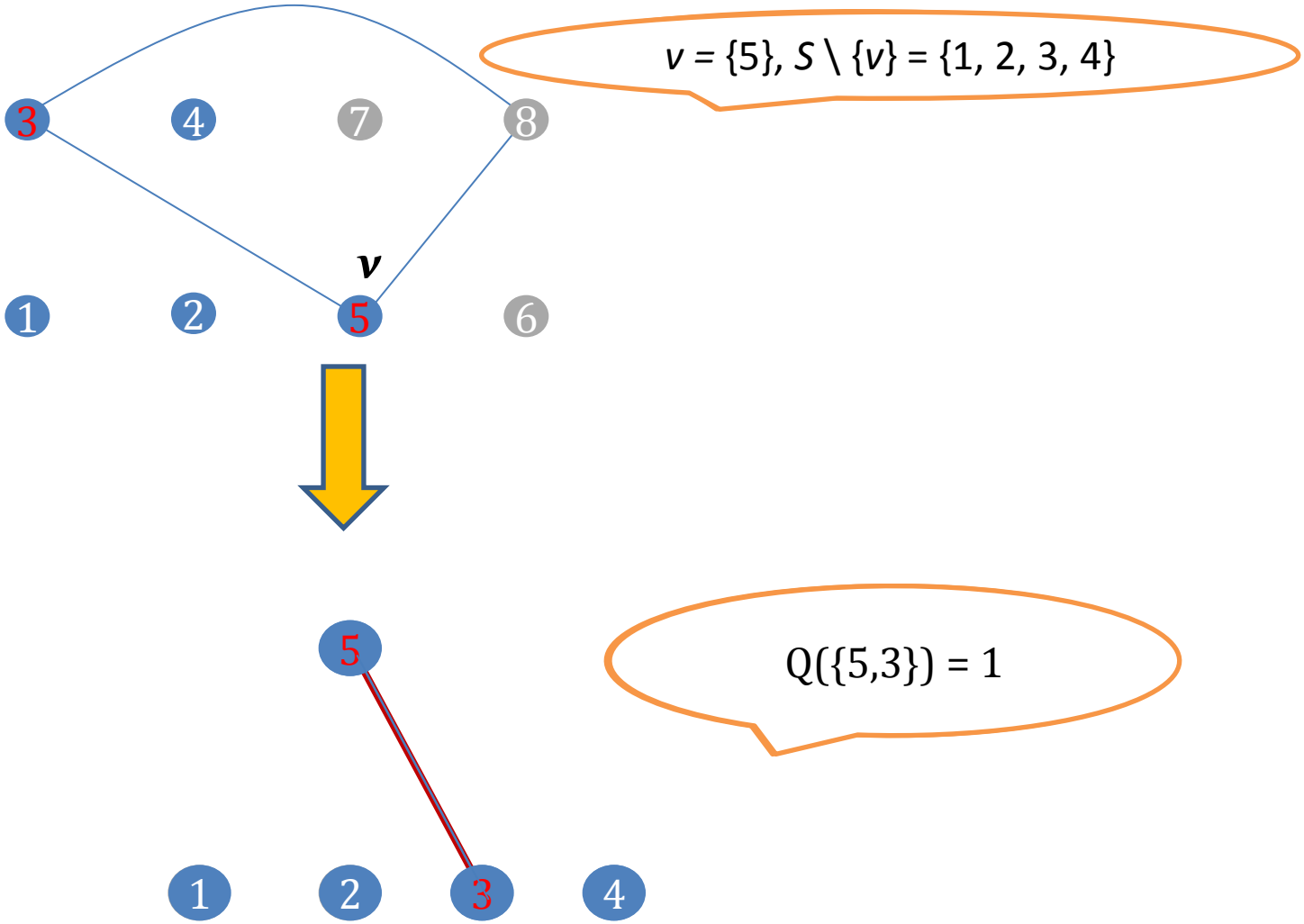



Q({1,2,3,4}) = 0
3
4
7
8
1
2
5
6

# The vertex is 5 or 7



# After 5 is determined

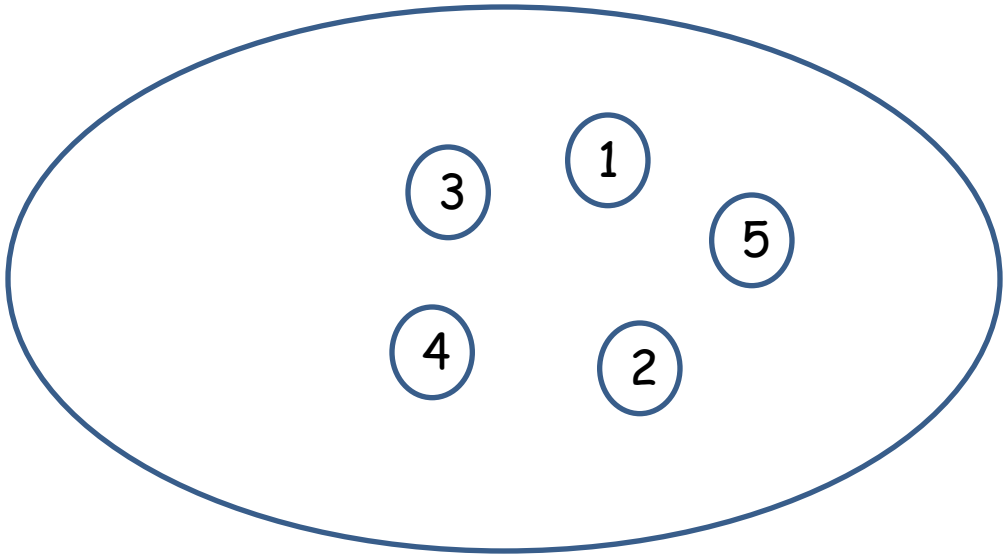


# The original work

- Delete the edge {3, 5} and keep searching for the others.
- We add one more trick to partition the vertex set independent subsets.
- So, all the edges are obtained from connecting vertices from different sets which is comparatively quicker since we keep some knowledge occurred in search!

$G = (V,E)$

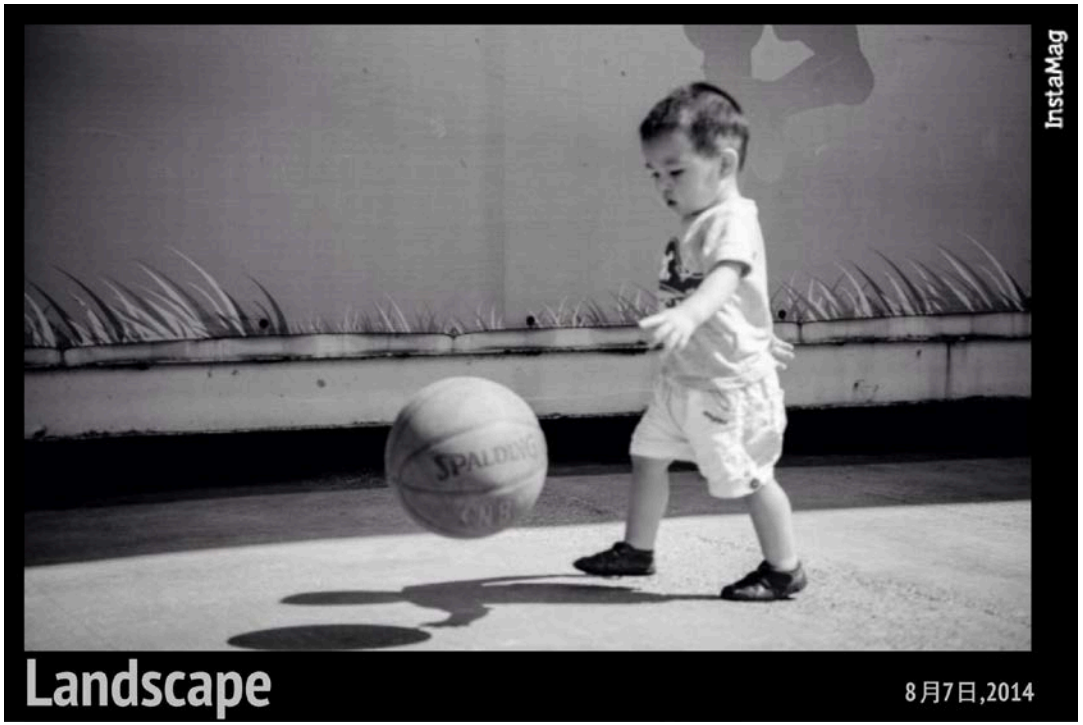
# Example of hidden hypergraph

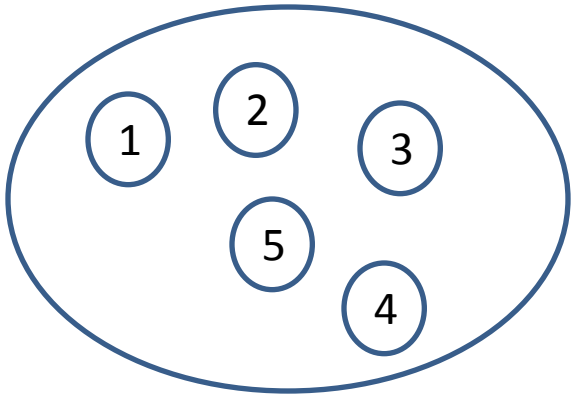



$V = \{1,2,3,4,5\}$

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

# *Step 1*

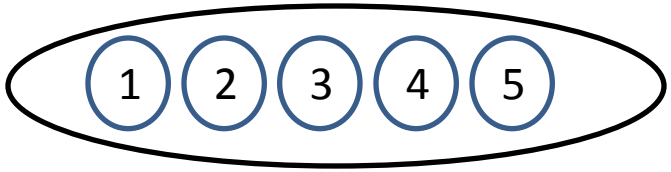





$V = \{1,2,3,4,5\}$

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

---



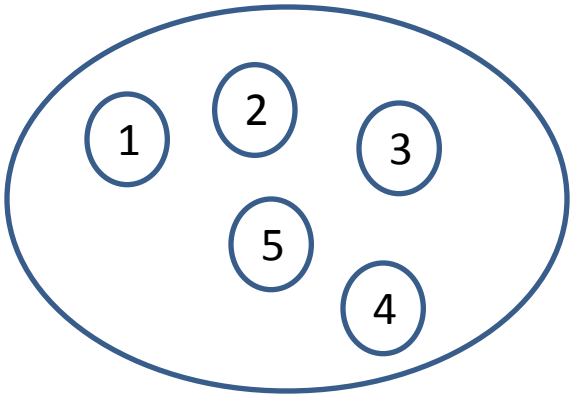

$S_1$

Initial setting:

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

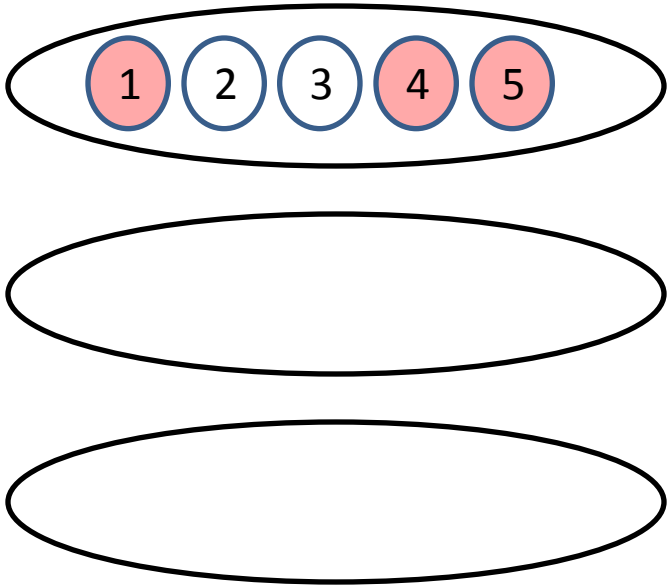
Find $\{1,4,5\}$

How?




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

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

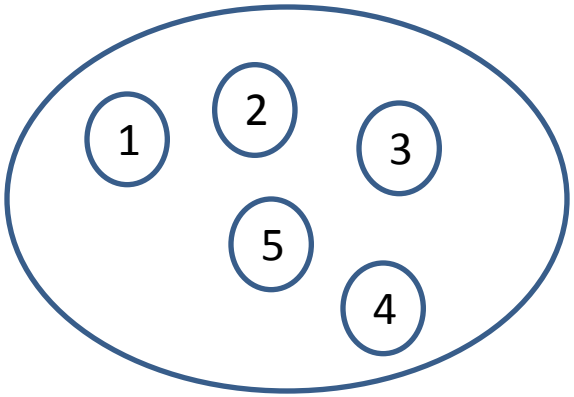



$S_0$

Initial setting:

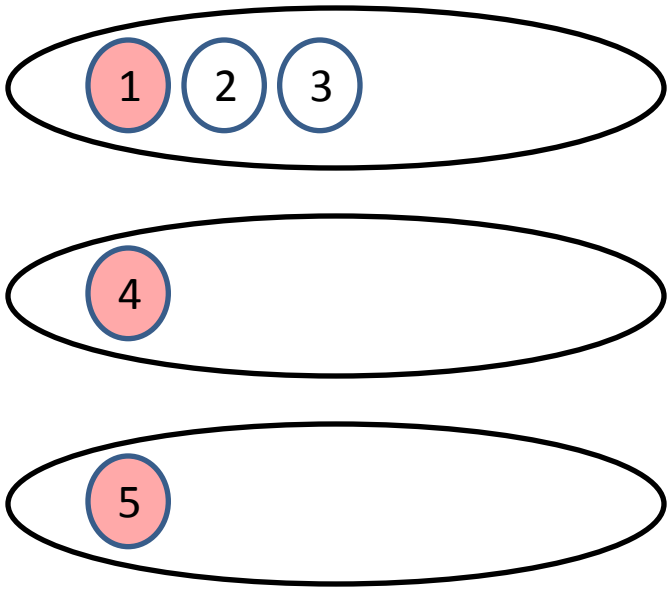
$S_0$ = {1,2,3,4,5}

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:

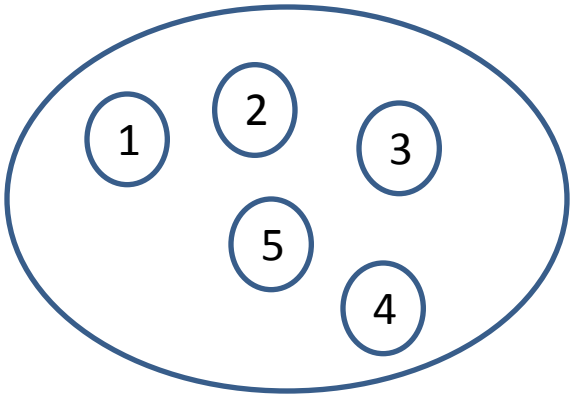
$S_1 = \{1,2,3,4,5\}$

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

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

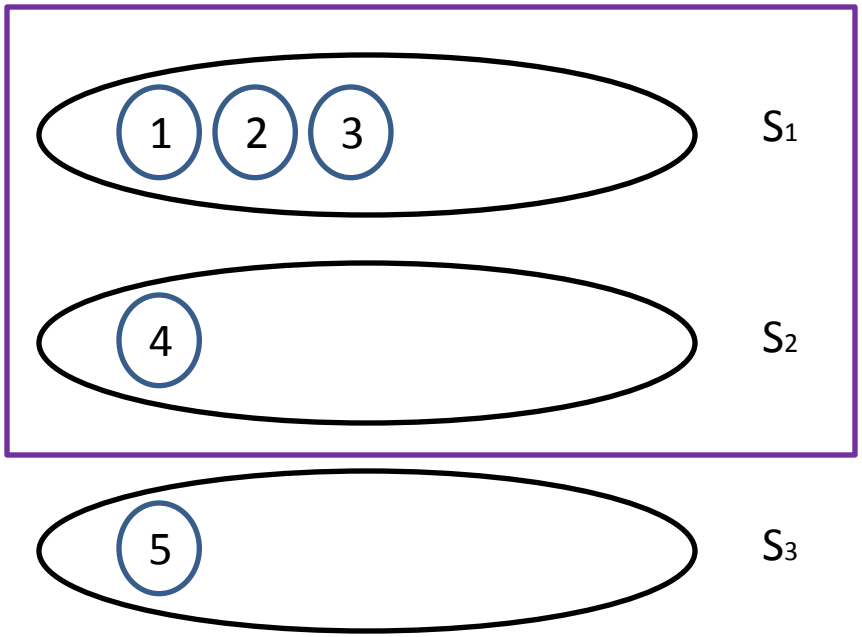
$S_2 = \{4\}$

$S_3 = \{5\}$




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

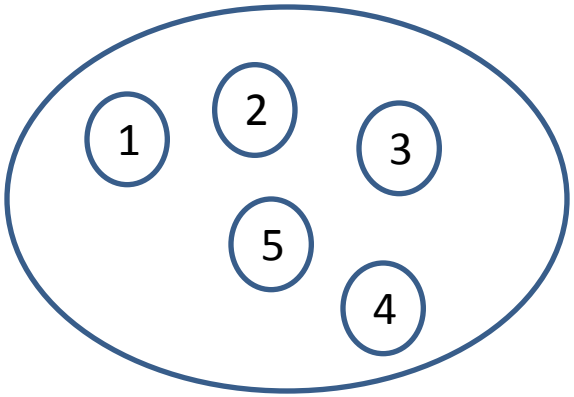
$E=\{\{1,2,3\},\{1,2,4\},\{1,3,4\},\{1,4,5\}\}$




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

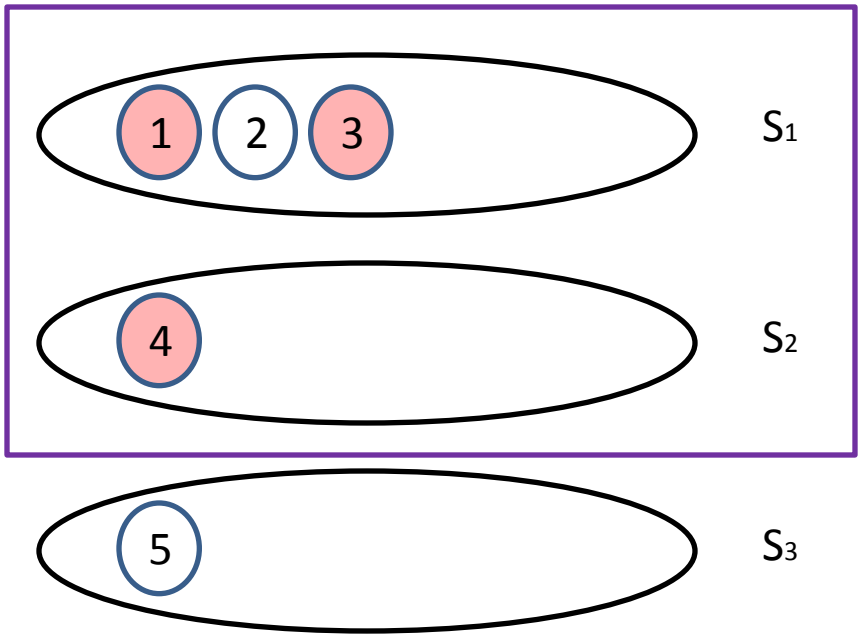
$S_2 = \{4\}$

$S_3 = \{5\}$




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

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

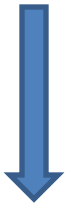



$S_1 = \{1,2,3\}$

$S_2 = \{4\}$

$S_3 = \{5\}$

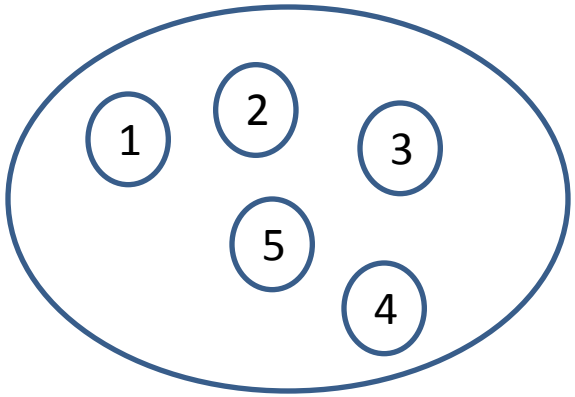
Find $\{1,3,4\}$



$S_1 = \{1,2\}$

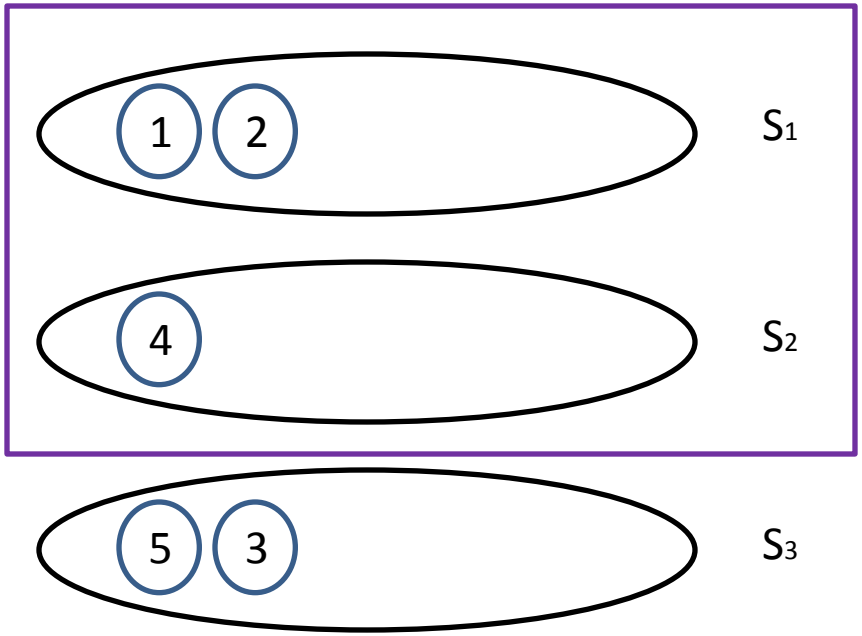
$S_2 = \{4\}$

$S_3 = \{3,5\}$




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

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

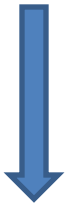



$S_1 = \{1,2\}$

$S_2 = \{4\}$

$S_3 = \{3,5\}$

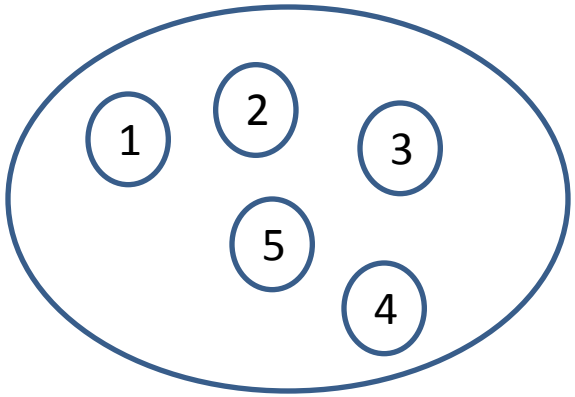
Find $\{1,2,4\}$



$S_1 = \{1\}$

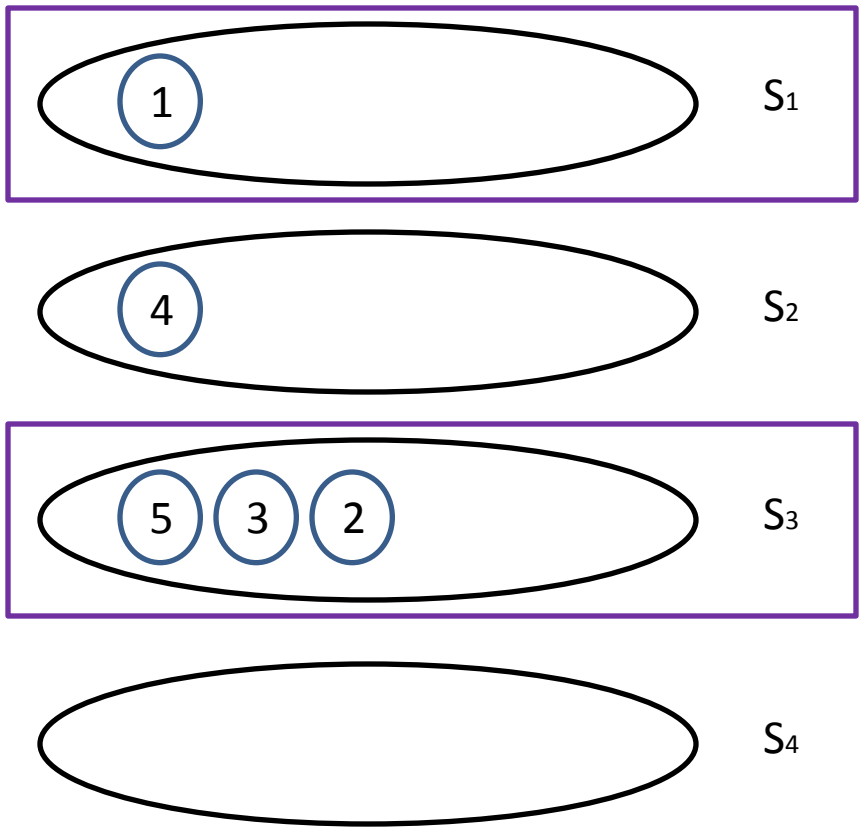
$S_2 = \{4\}$

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




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

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




$S_1 = \{1\}$

$S_2 = \{4\}$

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

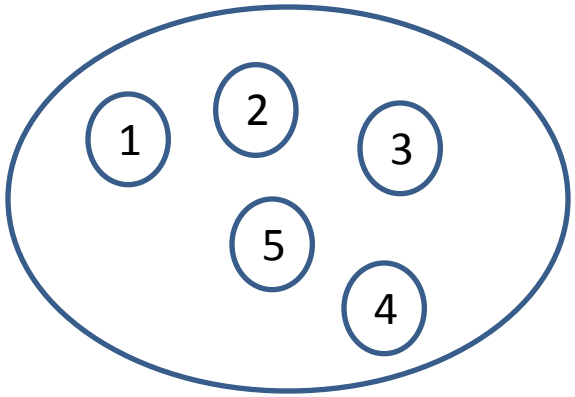
Find {1,2,3}

$S_1 = \{1\}$

$S_2 = \{4\}$

$S_3 = \{2,5\}$

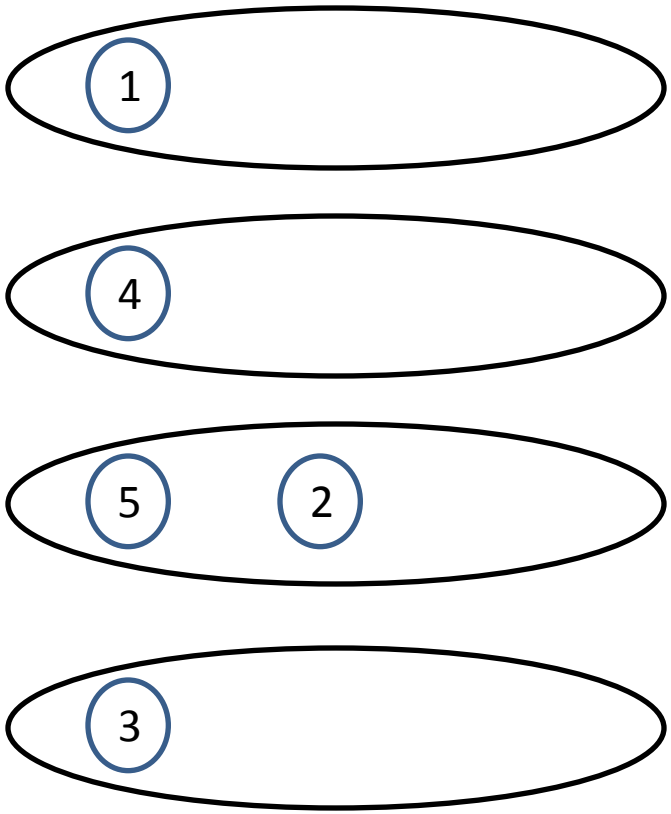
$S_4 = \{3\}$




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

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

---




$S_1$

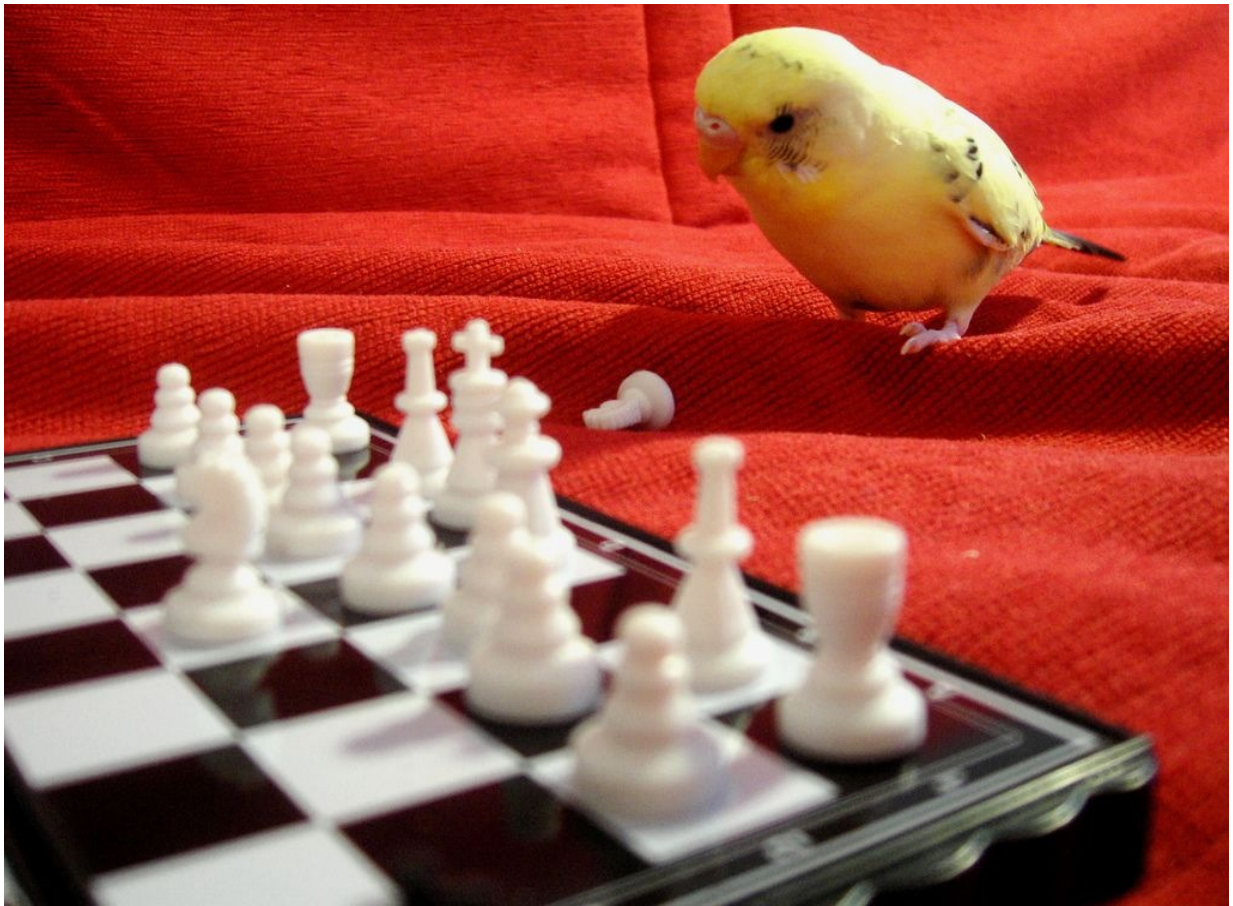
$S_2$

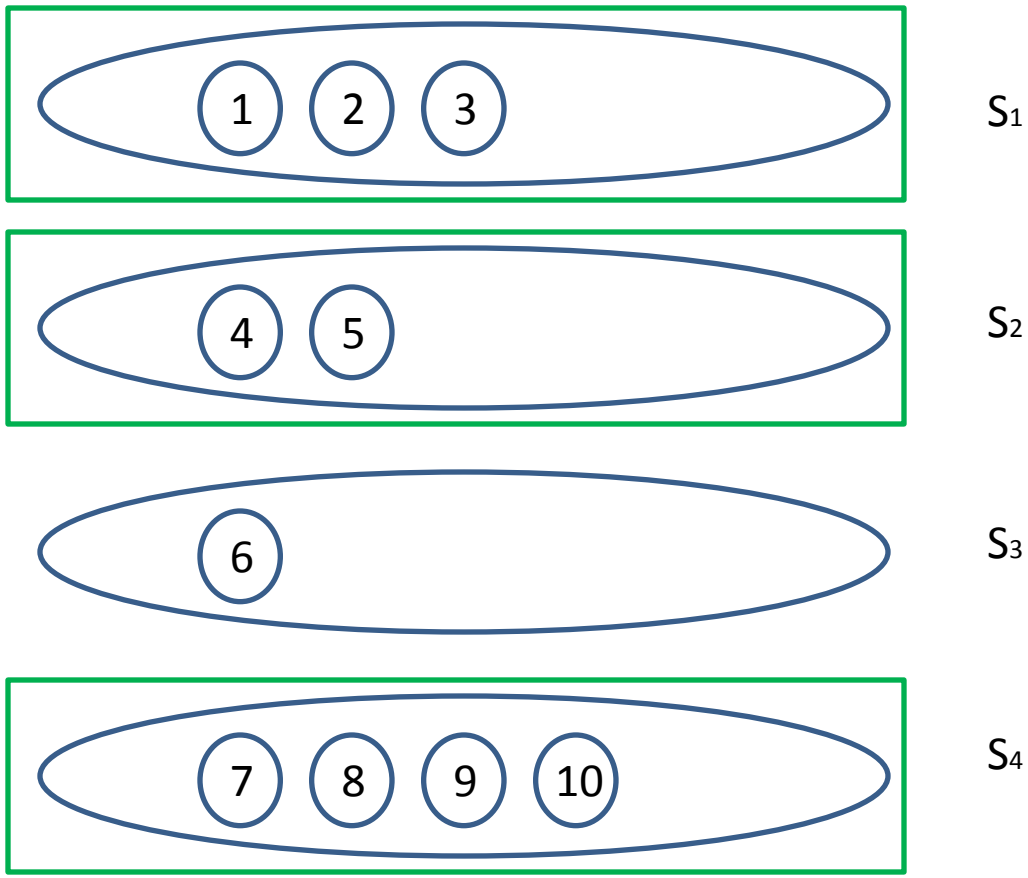
$S_3$

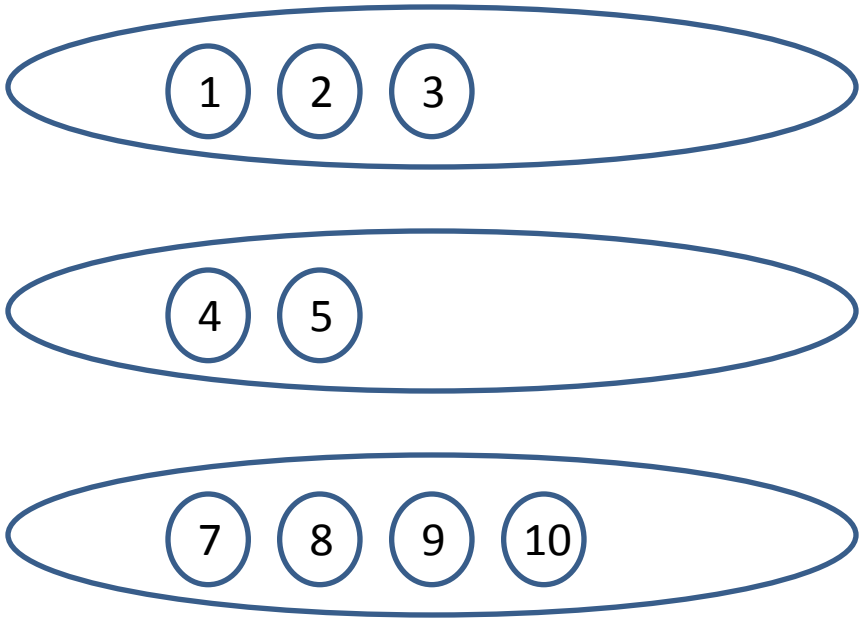
$S_4$

$S_1 = \{1\}$
$S_2 = \{4\}$
$S_3 = \{2,5\}$
$S_4 = \{3\}$

# *Step 2*




1
2
3
S1
4
5
S2
6
S3
7
8
9
10
S4



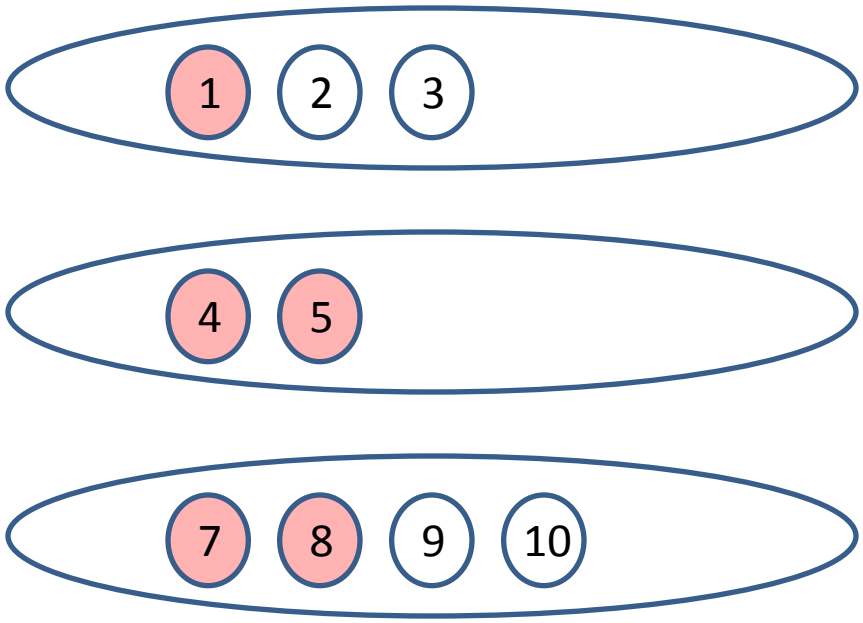

$S_1$

$S_2$

$S_4$

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

hidden




$S_1$

$S_2$

$S_4$

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

**NO**

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

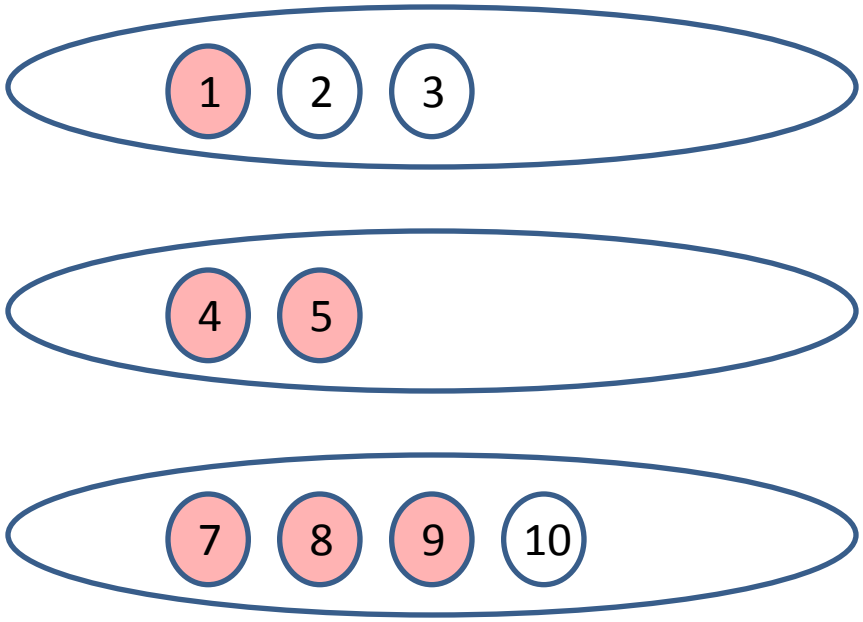
**NO**

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

**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**

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

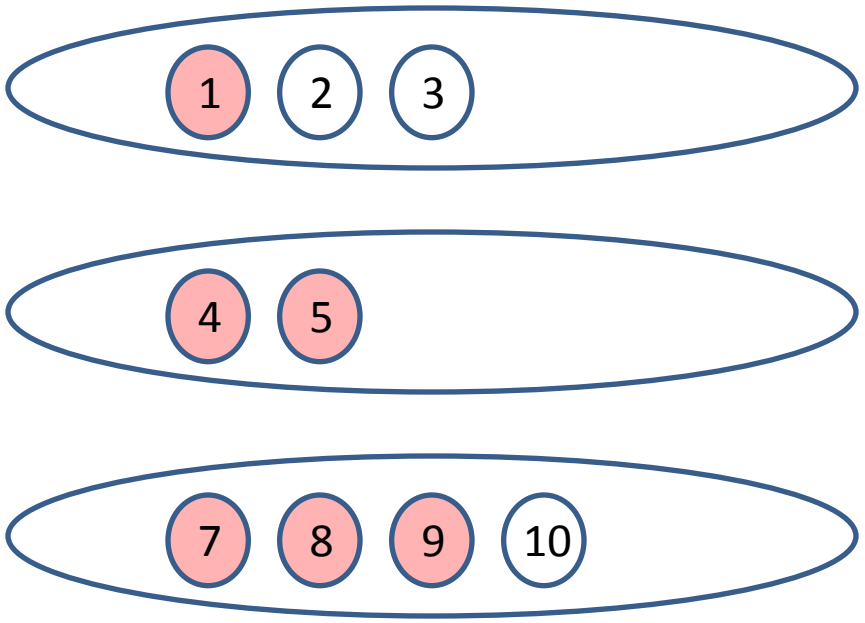
**NO**

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

**NO**

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

hidden




$S_1$

$S_2$

$S_4$

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

**NO**

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

**Yes**

$Q(\{1\} \cup \{5\} \cup \{7\})$ **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.
- It is by an *adaptive algorithm as you can see from the above example*.

# *References*


- A new construction of 3-bar-*separable* matrices via improved decoding of Macula construction (with F. K. Hwang), **Discrete Optim.**, 5(2008), 700-704.
- An upper bound of the number of tests in pooling designs for the *error-tolerant complex model* (with Hong-Bin. Chen and F. K. Hwang), **Optim. Lett.**, 2(2008), no. 3, 425-431.
- The minimum number of e-vertex-cover among *hypergraphs* with e edges of given rank (with F. H. Chang, F. K. Hwang and B. C. Lin), **Discrete Applied Math**., 157(2009), 164-169.
- Non-adaptive algorithms for *threshold* group testing (with Hong-Bin Chen), **Discrete Applied Math.**, 157(2009), 1581-1585.

# *Continued*

- 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.

# *Continued*

- Learning a *hidden graph* (with Huilan Chang and Chie-Huai Shih), **Optim. Lett.**, 8(2014), no. 8, 2341 – 2348.
- New bound on *2-bar-separable* codes of length 2 (with Minquan Cheng, J. Jiang, Yuan-Hsun Lo and Ying Miao), **Des. Code Cryptography**, 74(2015), no. 1, 31 – 40.
- Codes with the identifiable parent property for multimedia fingerprinting (with Minquan Cheng, J. Jiang, Yuan-Hsun Lo and Ying Miao), **Des. Code Cryptography**, 83(2017), no. 1, 71 – 82.
- Learning a hidden *uniform hypergraph* (with Huilan Chang and Chi-Huai Shih), **Optim. Lett**., 12(2018), 55 - 62.

# *Keep Moving Forward!*