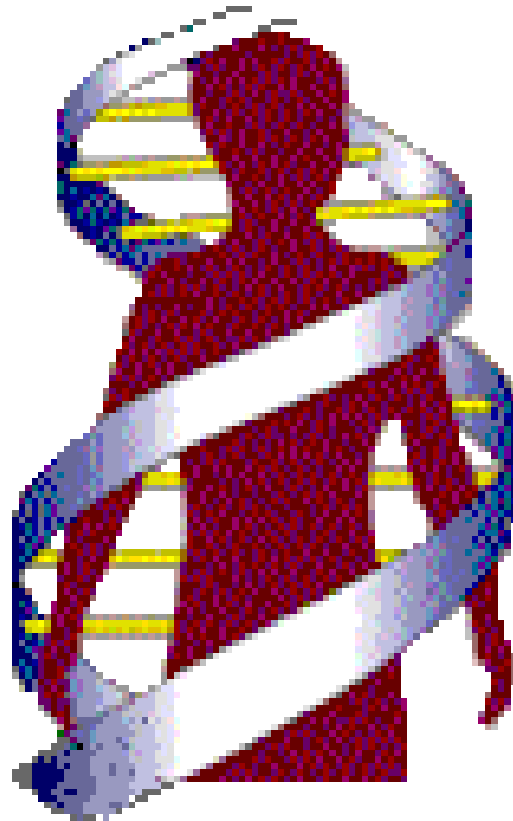


Group Testing – Applications

群試理論的應用

Hung-Lin Fu 傅 恆 霖

Human Genome Project



- The Human Genome Project was a 15-year-long, publicly funded project initiated in 1990 with the objective of determining the DNA sequence of the entire euchromatic human genome within 15 years.^[4] In May 1985, Robert Sinsheimer organized a workshop to discuss sequencing the human genome,^[5] but for a number of reasons the NIH was uninterested in pursuing the proposal. The following March, the Santa Fe Workshop was organized by [Charles DeLisi](#) and David Smith of the Department of Energy's Office of Health and Environmental Research (OHER).^[6] At the same time [Renato Dulbecco](#) proposed whole genome sequencing in an essay in Science.^[7] James Watson followed two months later with a workshop held at the Cold Spring Harbor Laboratory.

- The fact that the Santa Fe workshop was motivated and supported by a Federal Agency opened a path, albeit a difficult and tortuous one,^[8] for converting the idea into public policy. In a memo to the Assistant Secretary for Energy Research (Alvin Trivelpiece), Charles DeLisi, who was then Director of the OHER, outlined a broad plan for the project.^[9]

- This started a long and complex chain of events which led to approved reprogramming of funds that enabled the OHER to launch the Project in 1986, and to recommend the first line item for the HGP, which was in President Reagan's 1988 budget submission,^[8] and ultimately approved by the Congress. Of particular importance in Congressional approval was the advocacy of Senator [Peter Domenici](#), whom DeLisi had befriended.^[10] Domenici chaired the Senate Committee on Energy and Natural Resources, as well as the Budget Committee, both of which were key in the DOE budget process. Congress added a comparable amount to the NIH budget, thereby beginning official funding by both agencies.

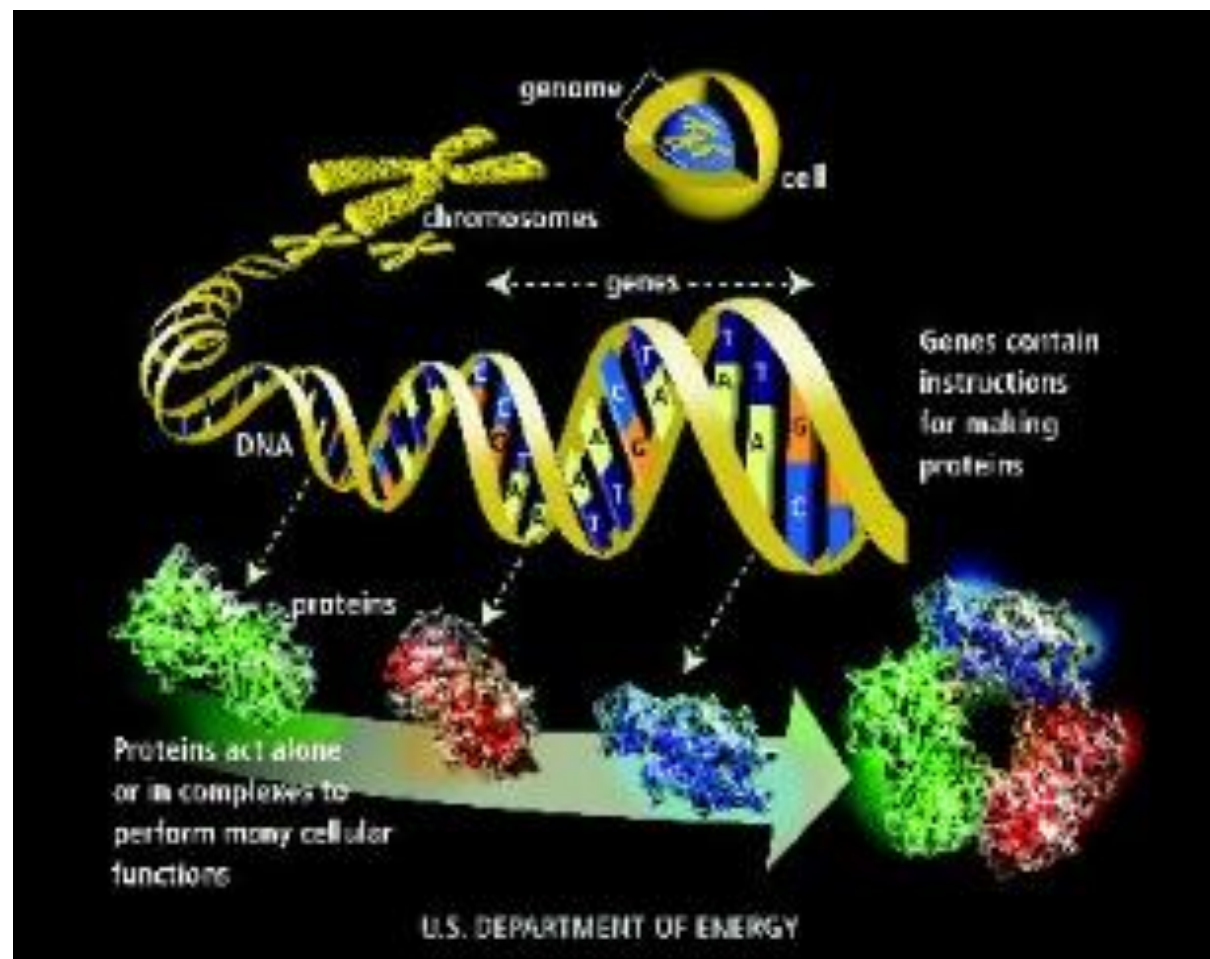
- [Alvin Trivelpiece](#) sought and obtained the approval of DeLisi's proposal by Deputy Secretary [William Flynn Martin](#). This chart^[11] was used in the spring of 1986 by Trivelpiece, then Director of the Office of Energy Research in the Department of Energy, to brief Martin and Under Secretary Joseph Salgado regarding his intention to reprogram \$4 million to initiate the project with the approval of [Secretary Herrington](#). This reprogramming was followed by a line item budget of \$16 million in the [Reagan Administration](#)'s 1987 budget submission to Congress.^[12] It subsequently passed both Houses. The Project was planned for 15 years.^[13]
- Candidate technologies were already being considered for the proposed undertaking at least as early as 1985.^[14]

- In 1990, the two major funding agencies, DOE and [NIH](#), developed a memorandum of understanding in order to coordinate plans and set the clock for the initiation of the Project to 1990.^[15] At that time, David Galas was Director of the renamed “Office of Biological and Environmental Research” in the U.S. Department of Energy’s Office of Science and [James Watson](#) headed the NIH Genome Program. In 1993, Aristides Patrinos succeeded Galas and [Francis Collins](#) succeeded [James Watson](#), assuming the role of overall Project Head as Director of the [U.S. National Institutes of Health](#) (NIH) National Center for Human Genome Research (which would later become the [National Human Genome Research Institute](#)). A working draft of the genome was announced in 2000 and the papers describing it were published in February 2001. A more complete draft was published in 2003, and genome "finishing" work continued for more than a decade

- The \$3-billion project was formally founded in 1990 by the US Department of Energy and the National Institutes of Health, and was expected to take 15 years.^[16] In addition to the United States, the international [consortium](#) comprised [geneticists](#) in the United Kingdom, France, Australia, China and myriad other spontaneous relationships.^[17]
- Due to widespread international cooperation and advances in the field of [genomics](#) (especially in [sequence analysis](#)), as well as major advances in computing technology, a 'rough draft' of the genome was finished in 2000 (announced jointly by U.S. President [Bill Clinton](#) and the British [Prime Minister Tony Blair](#) on June 26, 2000).^[18] This first available rough draft [assembly](#) of the genome was completed by the Genome Bioinformatics Group at the [University of California, Santa Cruz](#), primarily led by then graduate student [Jim Kent](#). Ongoing [sequencing](#) led to the announcement of the essentially complete [genome](#) on April 14, 2003, two years earlier than planned.^{[19][20]} In May 2006, another milestone was passed on the way to completion of the project, when the sequence of the [last chromosome](#) was published in [Nature](#).^[21]

How does the human genome stack up?

Organism	Genome Size (Bases)	Estimated Genes
Human (人類)	3 billion	30,000
Laboratory mouse (白老鼠)	2.6 billion	30,000
Mustard weed (<i>A. thaliana</i>)	100 million	25,000
Roundworm (<i>C. elegans</i>)	97 million	19,000
Fruit fly (果蠅)	137 million	13,000
Yeast (酵母菌)	12.1 million	6,000
Bacterium (大腸桿菌)	4.6 million	3,200
Human immunodeficiency virus (HIV)	9700	9



- DNA: Deoxyribonucleic Acid
- Nucleotide (核苷酸)
 - Adenine (A)
 - Thymine (T)
 - Guanine (G)
 - Cytosine (C)

Computational Molecular Biology

- DNA解序後，需分析訊息，但資料量30億對。
 - 需要現代科技輔助。

- 主要課題

- 序列組合
- 序列分析
- 基因認定
- 生物資訊資料庫、種族樹建構、蛋白質三維結構推測…。

序列比對探究

- 說明
 - 找序列中「相似」及「相異」的部份。
 - 為何要比對序列？
 - 為何要使用電腦比對？
 - 如何使用電腦比對？
 - 困難點：序列型態多樣、需建構不同的資料結構及**演算法** (*Algorithm*)。

Example of Group Testing

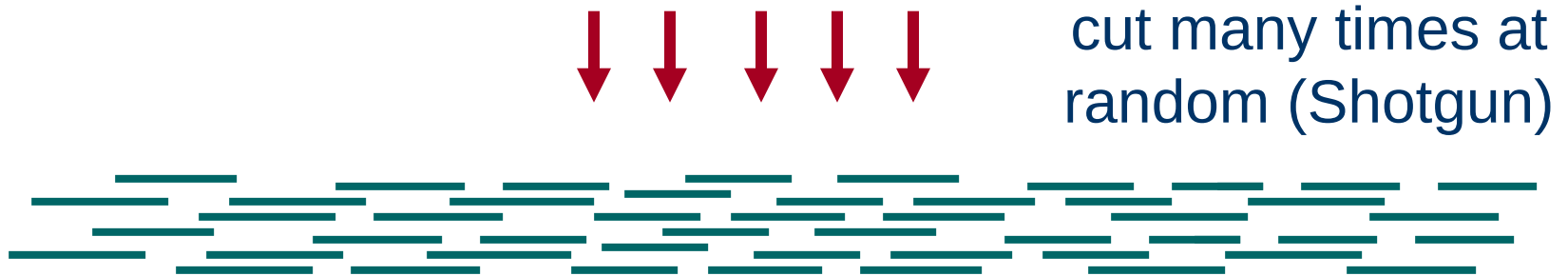
- In screening clone library the goal is to determine which clones in the clone library hybridize with a given probe in an efficient fashion. A clone is said to be **positive** if it hybridize with the **probe(探針)**, and **negative** otherwise.

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 projects, we are left with a collection of **contigs** that for special reasons cannot be assembled with general assembly algorithms. **Continued ...**

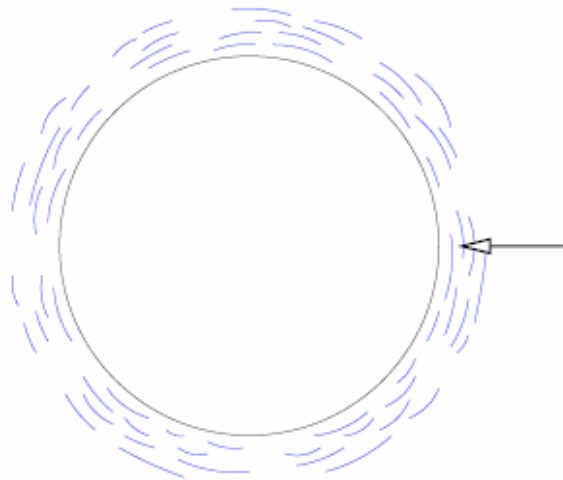
Random shotgun approach

genomic segment



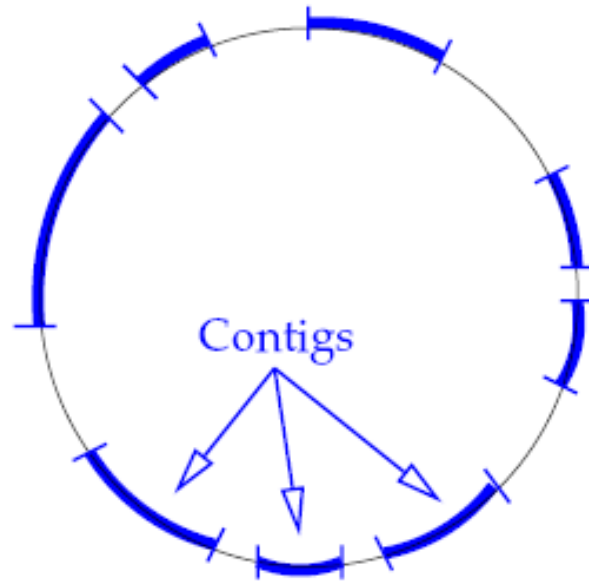
Whole-genome shotgun sequencing

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

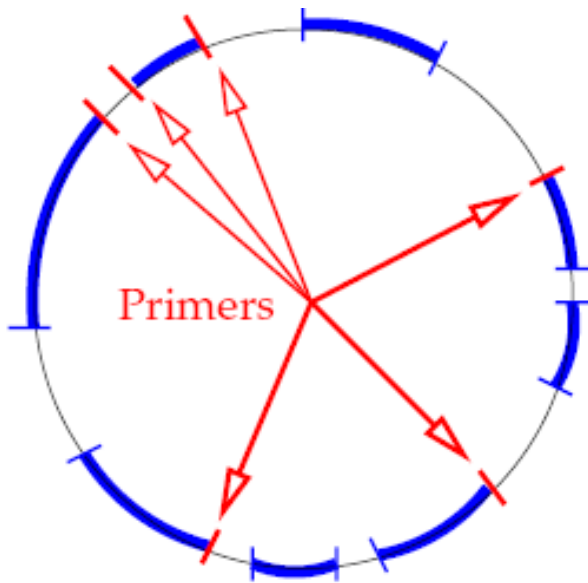


Circular genome

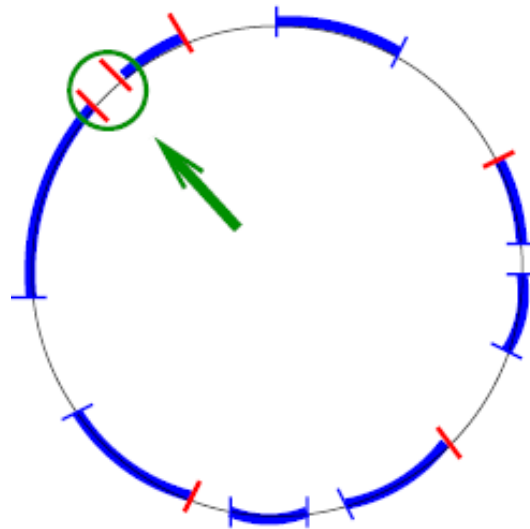
- Reads are assembled into contigs with unknown relative placement.

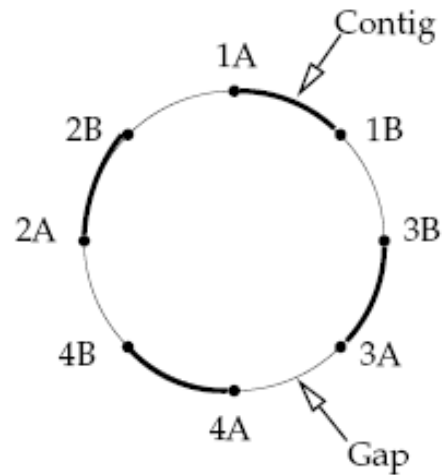


– **Primers** : (short) fragments of DNA characterizing ends of contigs.



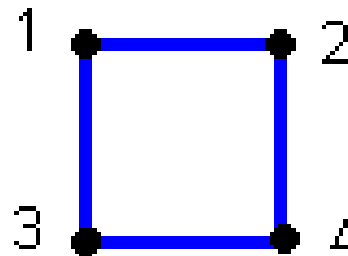
- A **PCR (Polymerase Chain Reaction)** reaction reveals if two primers are proximate (adjacent to the same gap).
- **Multiplex PCR** can treat multiple primers simultaneously and outputs if there is a pair of adjacent primers in the input set and even sometimes the number of such pairs.

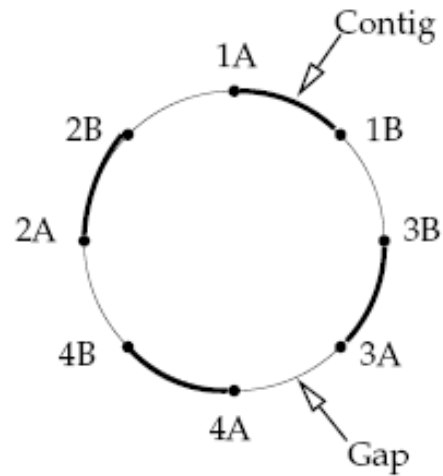




- Two primers of each contig are “mixed together”
 - Find a Hamiltonian cycle by PCRs!

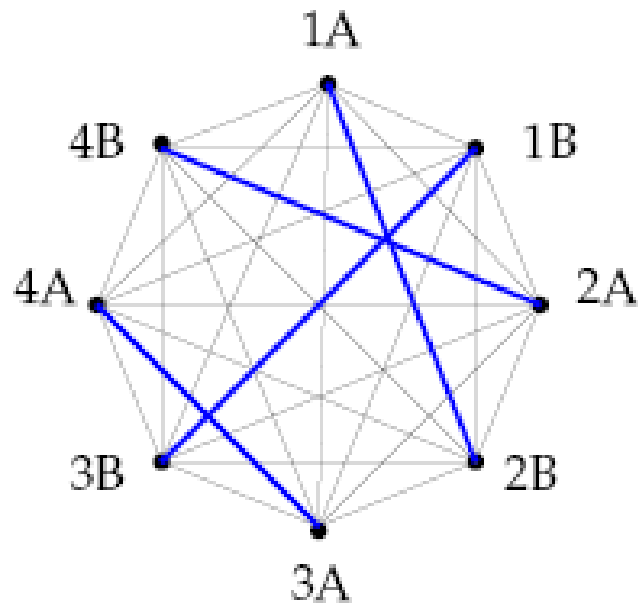
1A ——— 1B
 2A ——— 2B
 3A ——— 3B
 4A ——— 4B





1A ——— 1B
 2A ——— 2B
 3A ——— 3B
 4A ——— 4B

- 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 (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

Recent Result

- Theorem (Chang, Fu and Shih)

Let G be a graph with n vertices and m edges. Then we have an *adaptive algorithm* to reconstruct the hidden graph by using at most $m(2\log_2 n + 9)$ queries.

● Models

1) Multi-vertex model

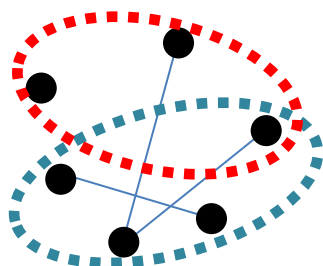
2) Quantitative multi-vertex model

3) k -vertex model

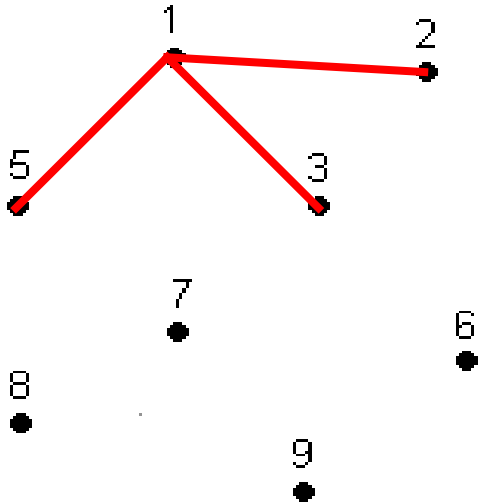
4) Quantitative k -multi-vertex model

Learning a hidden graph by edge-detecting queries:

$Q_G(S)$: does S include at least one edge of G ?



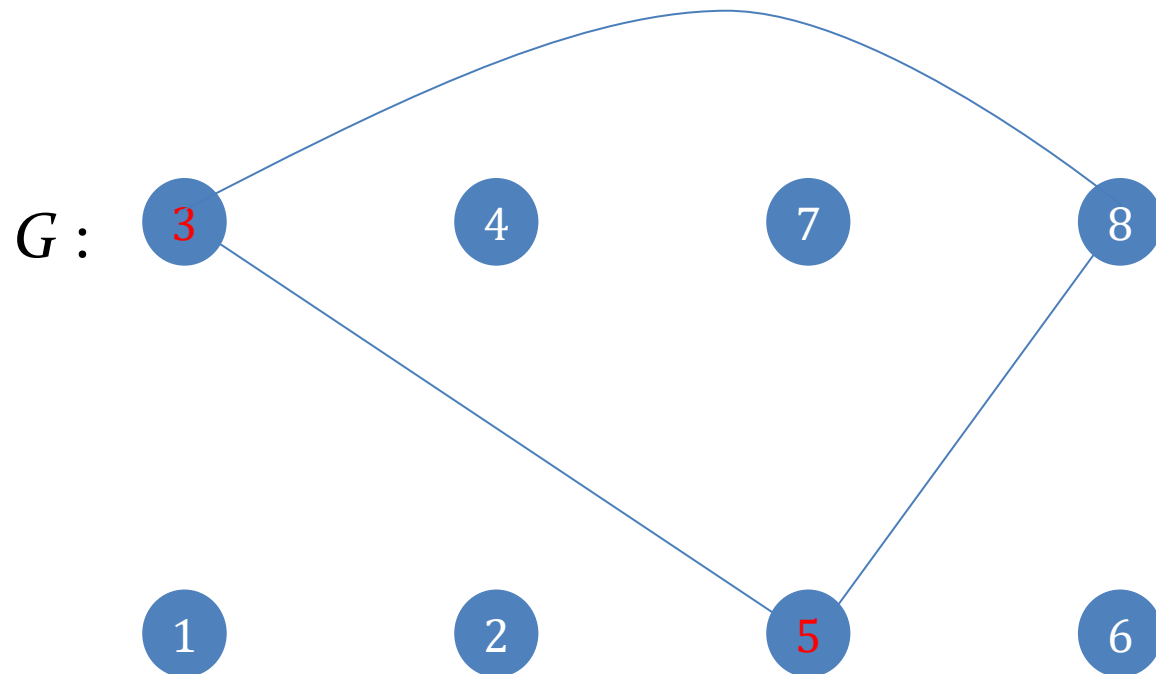
Hidden Graph

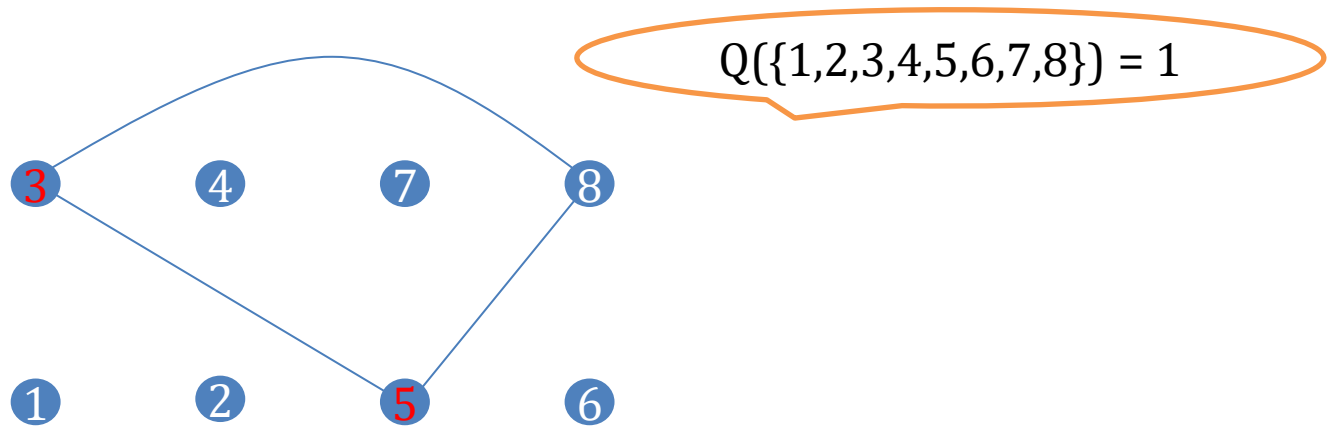


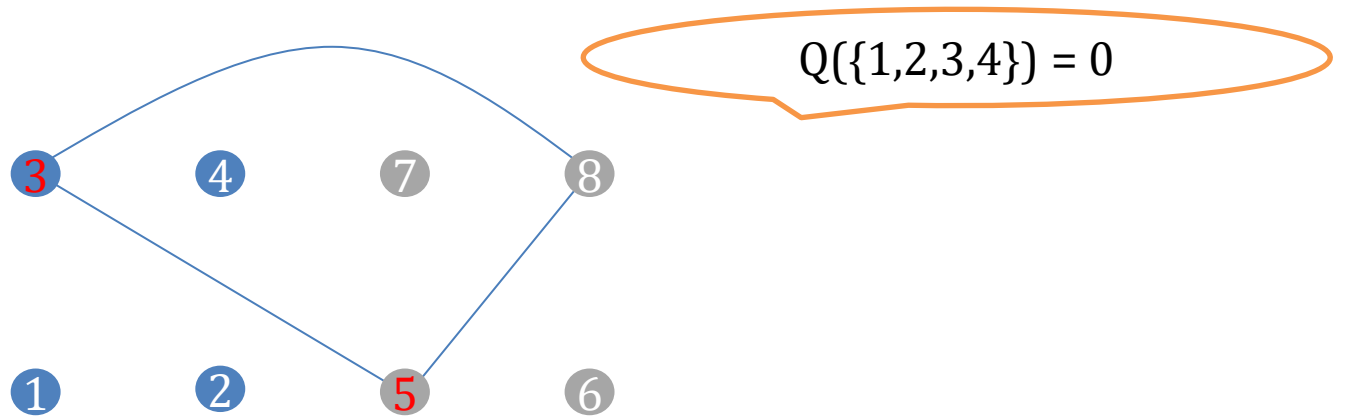
Algorithms

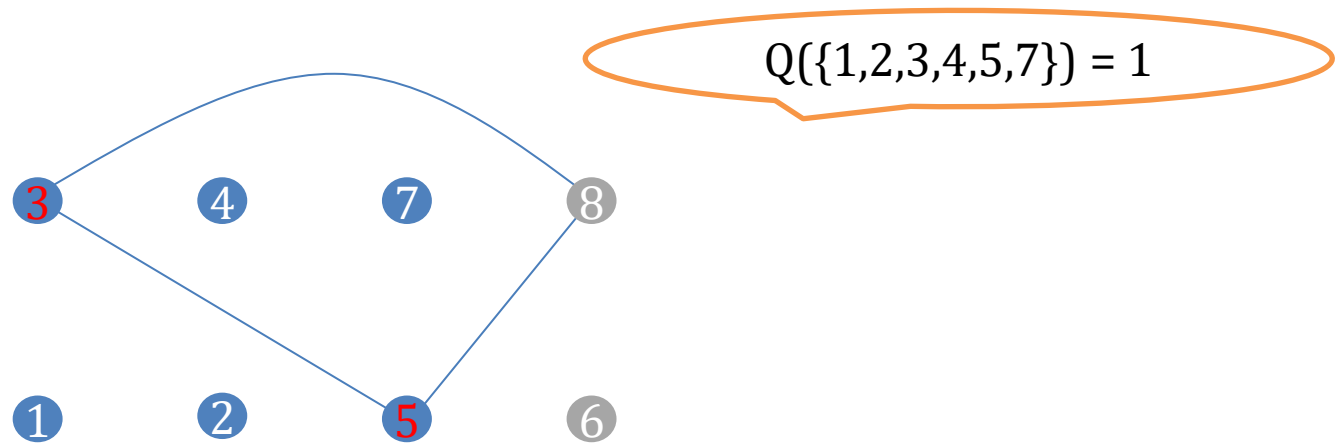
- *Adaptive algorithms:*
a query can depend on the answers obtained by previous queries.
- *Nonadaptive algorithms:*
queries are independent and can be processed in parallel.

Example

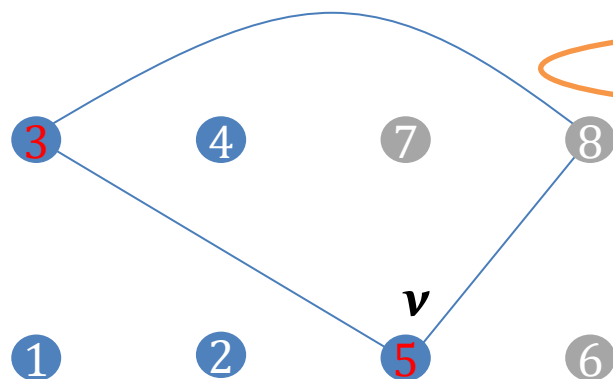




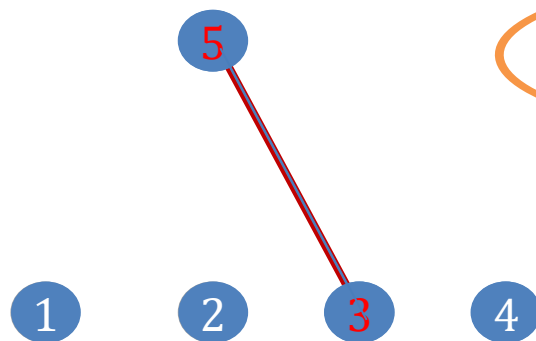




$$Q(\{1,2,3,4,5,7\}) = 1$$



$v = \{5\}, S \setminus \{v\} = \{1, 2, 3, 4\}$



$Q(\{5,3\}) = 1$

Hidden 3-uniform hypergraphs

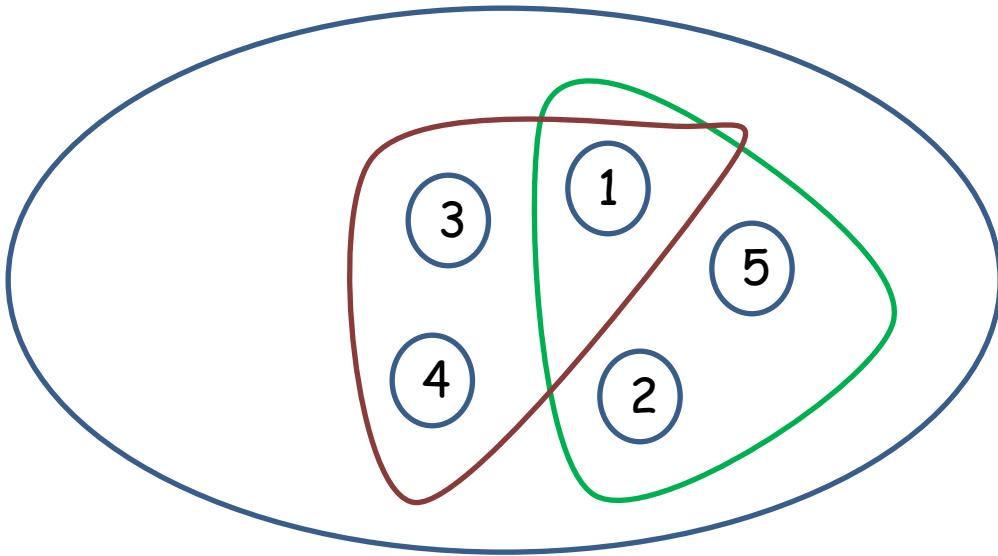
- The following example shows an idea of finding a hidden 3-uniform hypergraph.

Step 0:

Step 0: Don't Stop!



Find a hyperedge



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

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

Yes

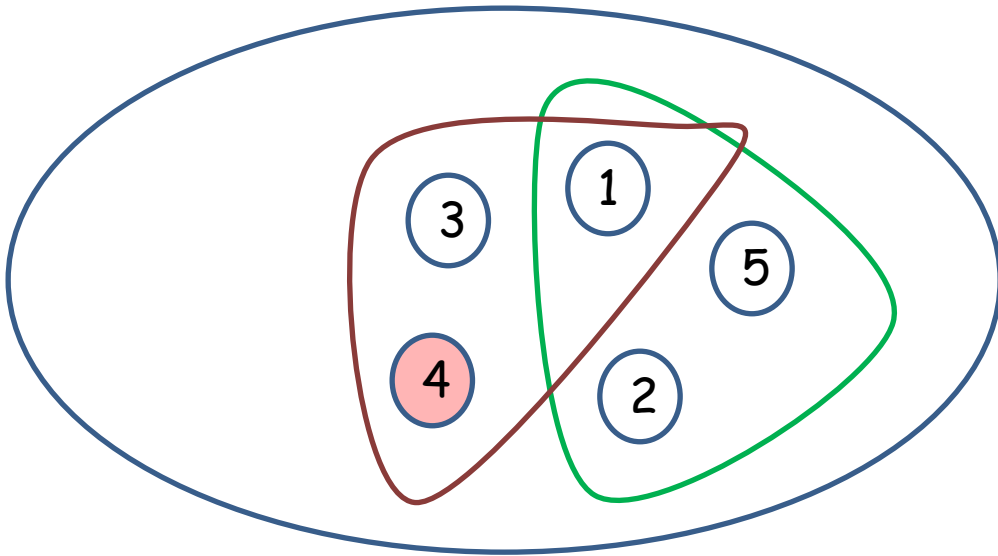
$Q(\{1, 2, 3\})$

No

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

Yes

Find a hyperedge



Algorithm

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

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

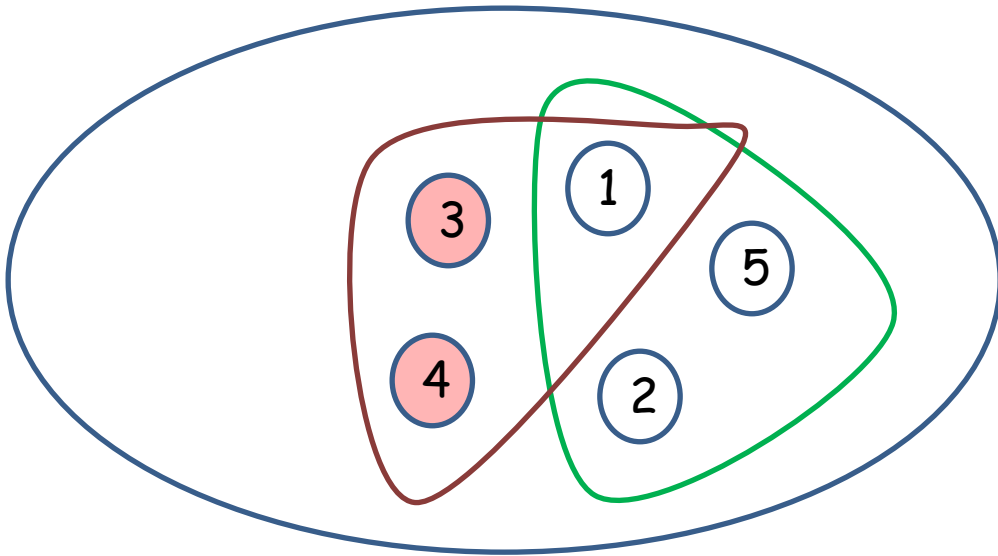
$Q(\{1, 2, 3\})$ No

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

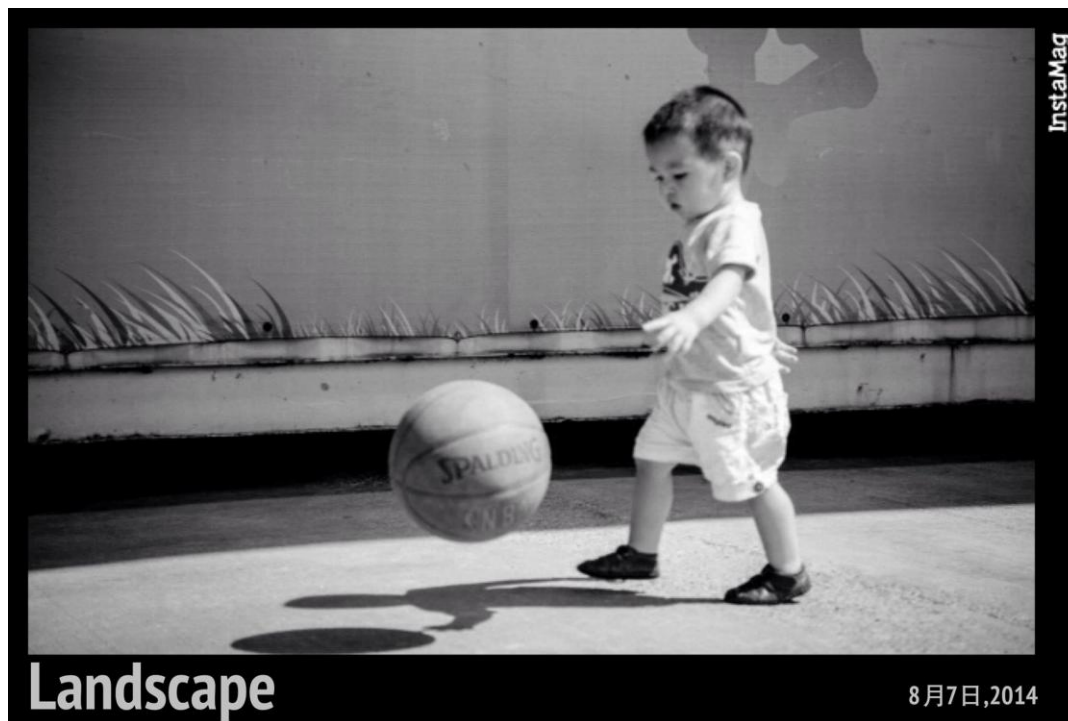
$\{4,1,2,3\}$

$Q(\{4,1,2\})$ No

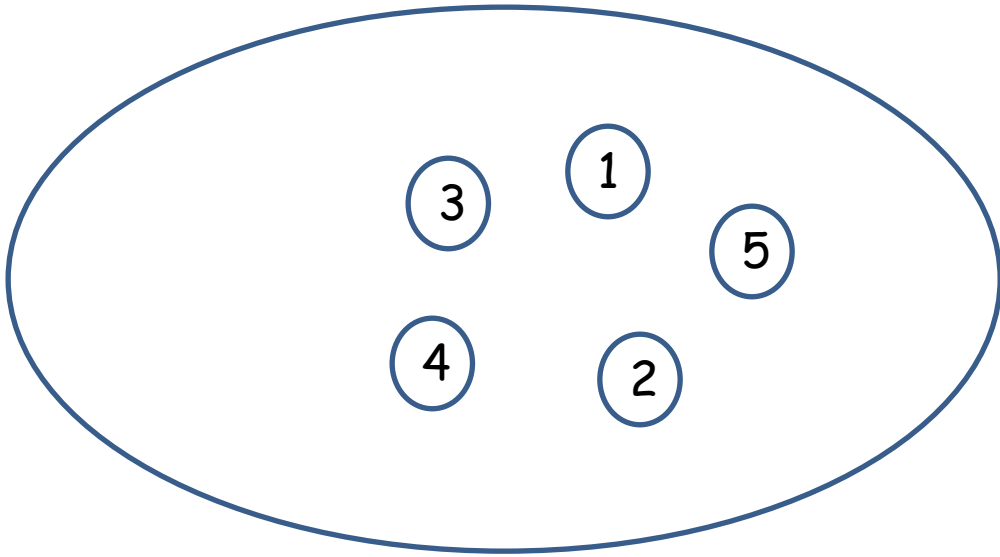
$\{4,3,1,2\}$

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

Step 1

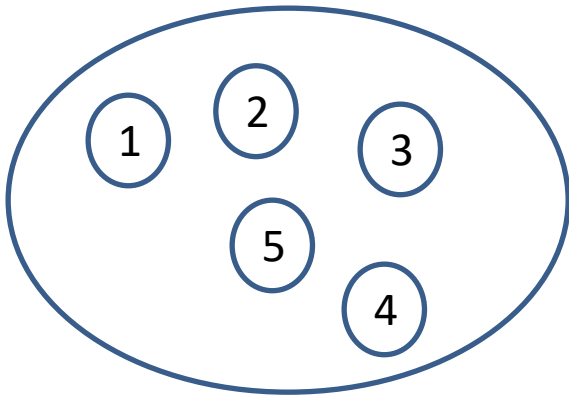


$$G=(V,E)$$



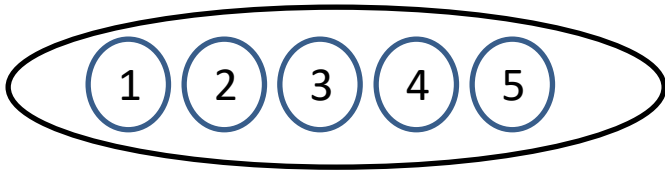
$$V=\{1,2,3,4,5\} \text{ (more edges)}$$

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



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

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

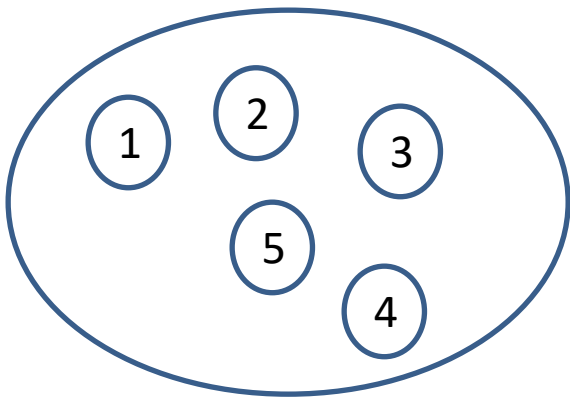


S_1

Initial setting:

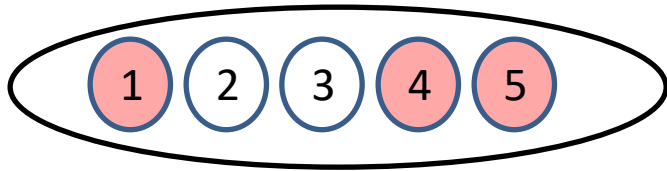
$S_1 = \{1, 2, 3, 4, 5\}$

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



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

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

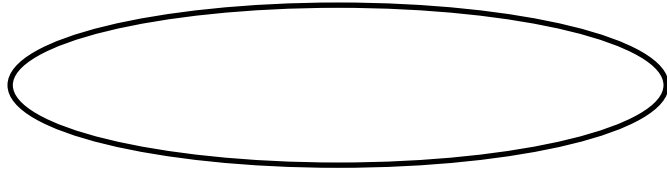
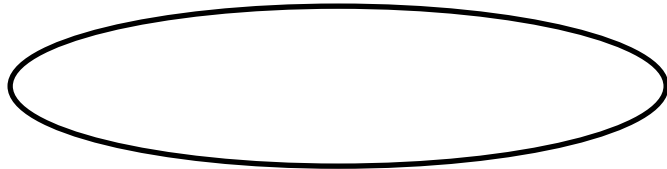


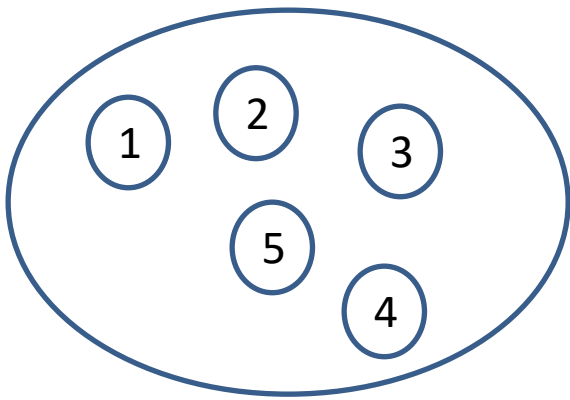
S_0

Initial setting:

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

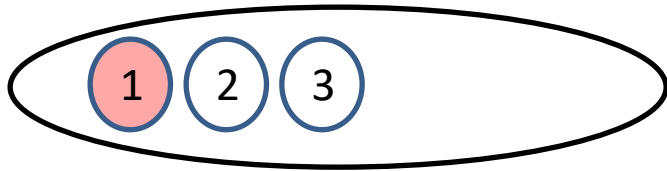
Find $\{1, 4, 5\}$



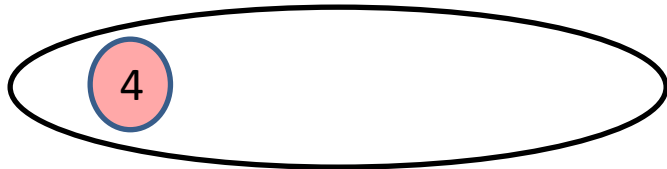


$V = \{1, 2, 3, 4, 5\}$

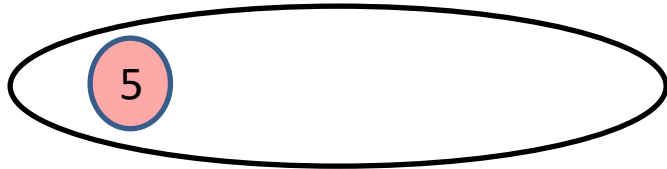
$E = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4, 5\}\}$



S_1



S_2

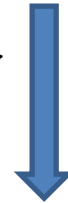


S_3

Initial setting:

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

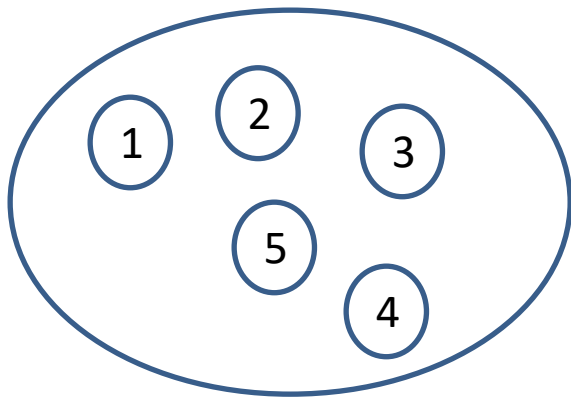
Find $\{1, 4, 5\}$



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

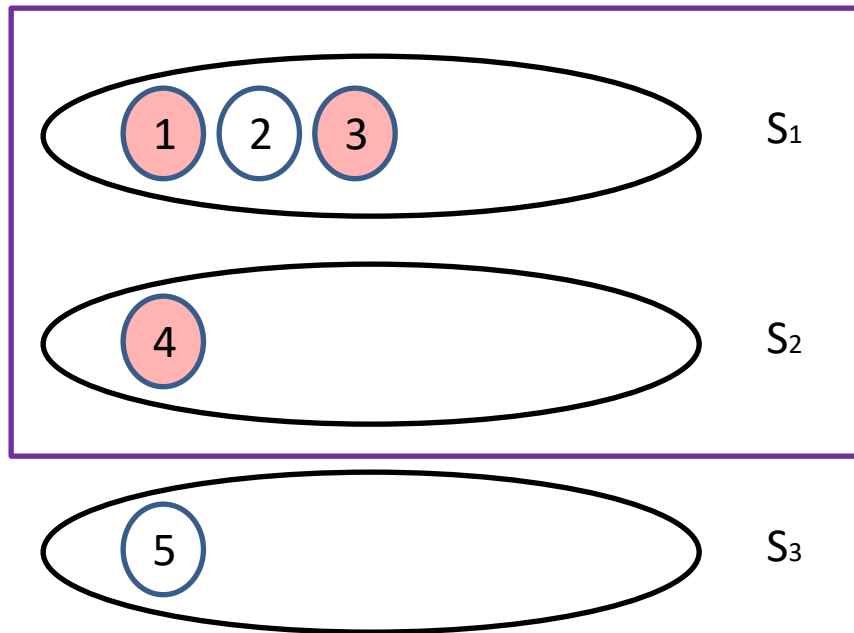
$S_2 = \{4\}$

$S_3 = \{5\}$



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

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



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

$S_2 = \{4\}$

$S_3 = \{5\}$

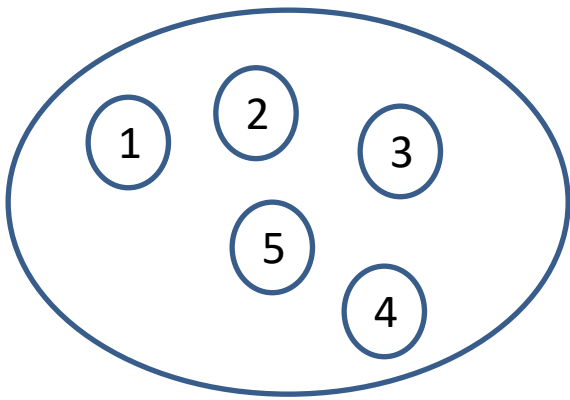
Find $\{1, 3, 4\}$



$S_1 = \{1, 2\}$

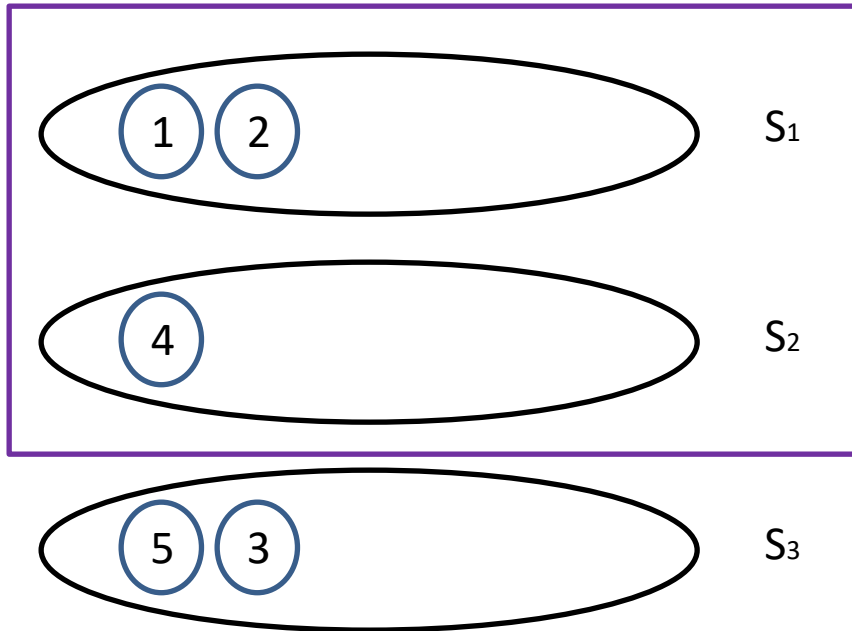
$S_2 = \{4\}$

$S_3 = \{3, 5\}$



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

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

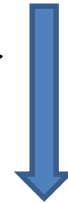


$S_1 = \{1, 2\}$

$S_2 = \{4\}$

$S_3 = \{3, 5\}$

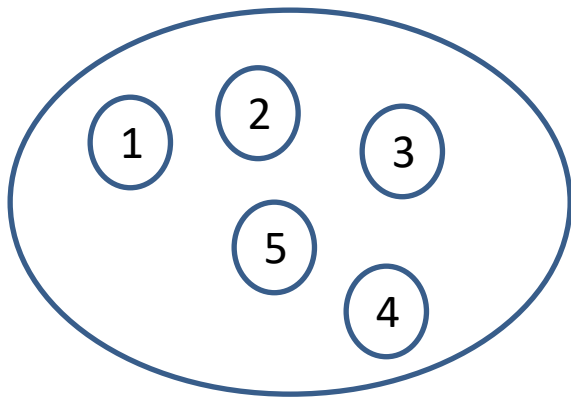
Find $\{1, 2, 4\}$



$S_1 = \{1\}$

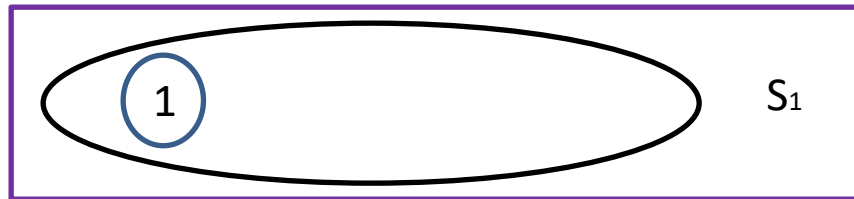
$S_2 = \{4\}$

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

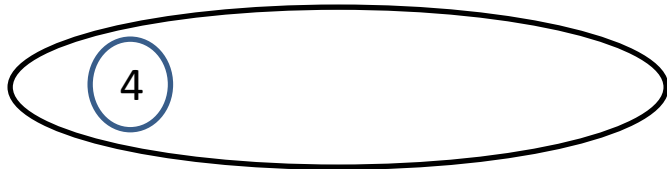


$V = \{1, 2, 3, 4, 5\}$

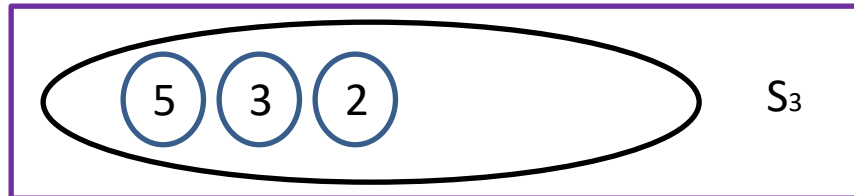
$E = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4, 5\}\}$



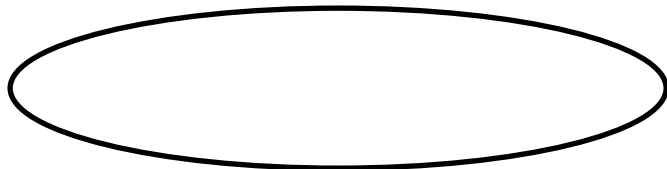
S_1



S_2



S_3



S_4

$S_1 = \{1\}$

$S_2 = \{4\}$

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

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

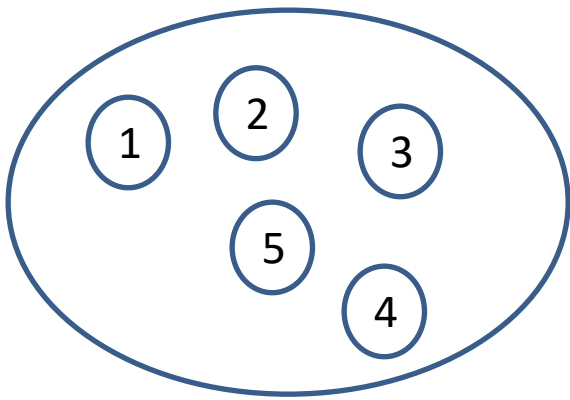


$S_1 = \{1\}$

$S_2 = \{4\}$

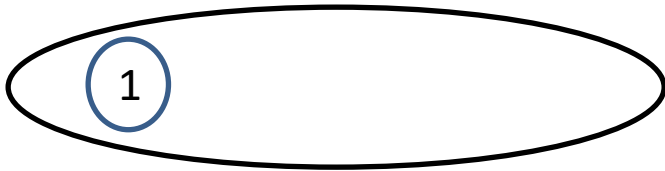
$S_3 = \{2, 5\}$

$S_4 = \{3\}$

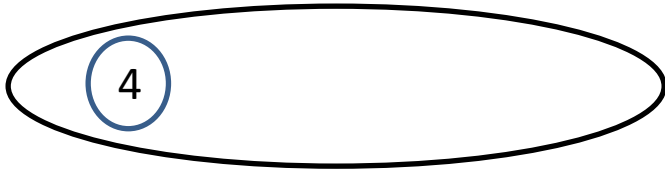


$V = \{1, 2, 3, 4, 5\}$

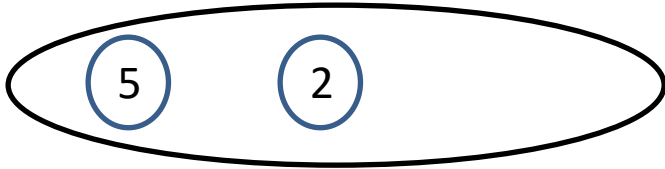
$E = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4, 5\}\}$



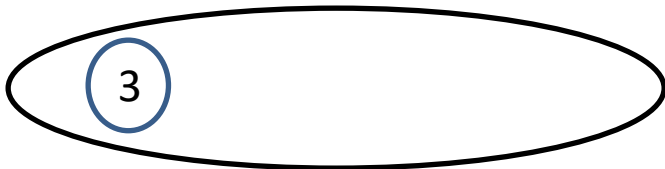
S_1



S_2



S_3



S_4

$S_1 = \{1\}$

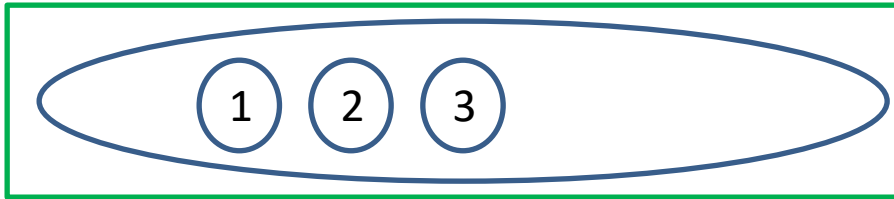
$S_2 = \{4\}$

$S_3 = \{2, 5\}$

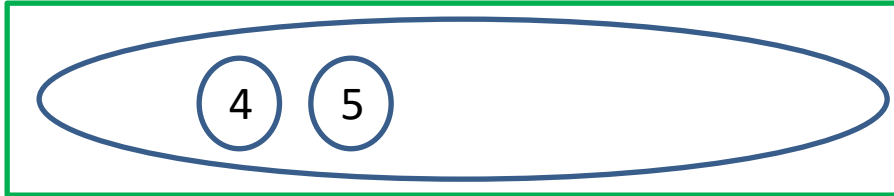
$S_4 = \{3\}$

Step 2

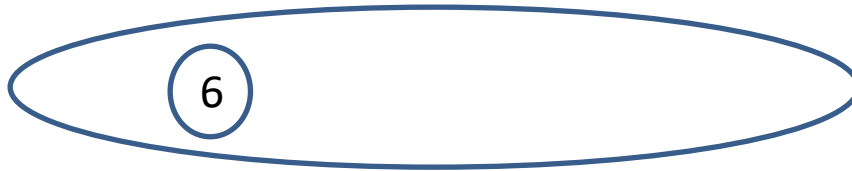




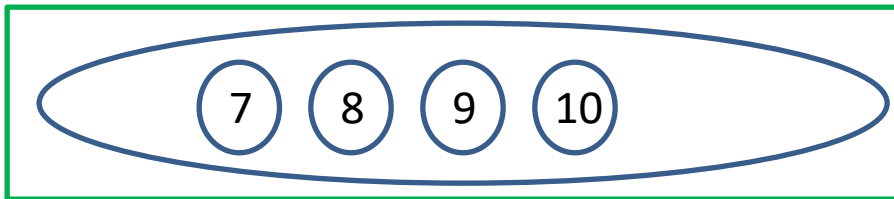
S_1



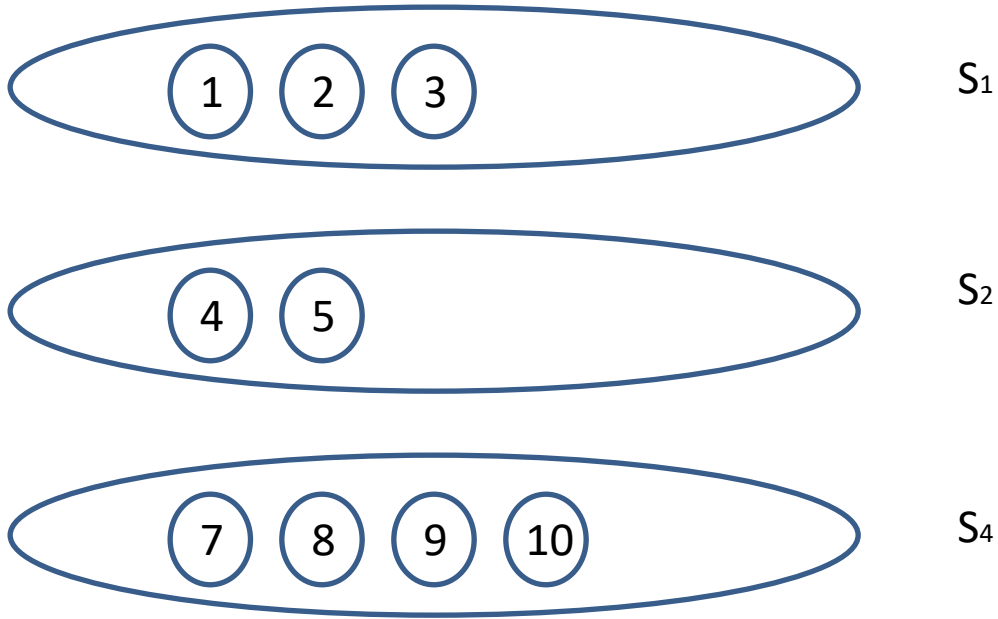
S_2



S_3

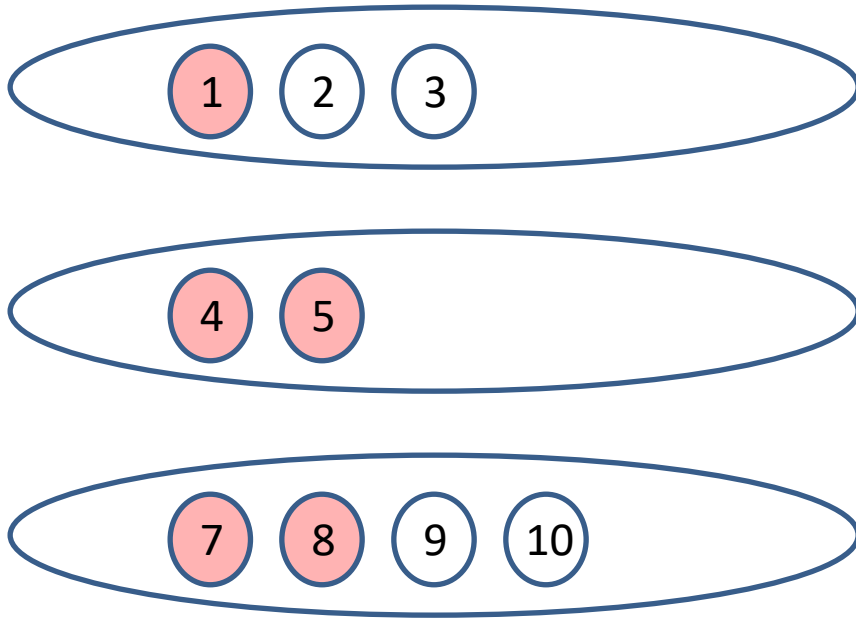


S_4



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

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

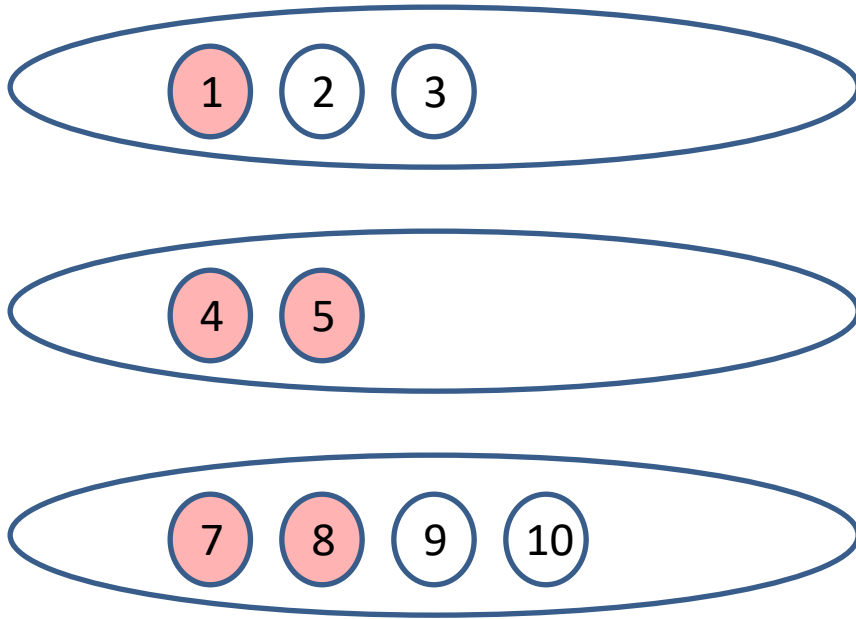
$$Q(S_1 \cup S_2 \cup S_4 - \{7, 8\})$$

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

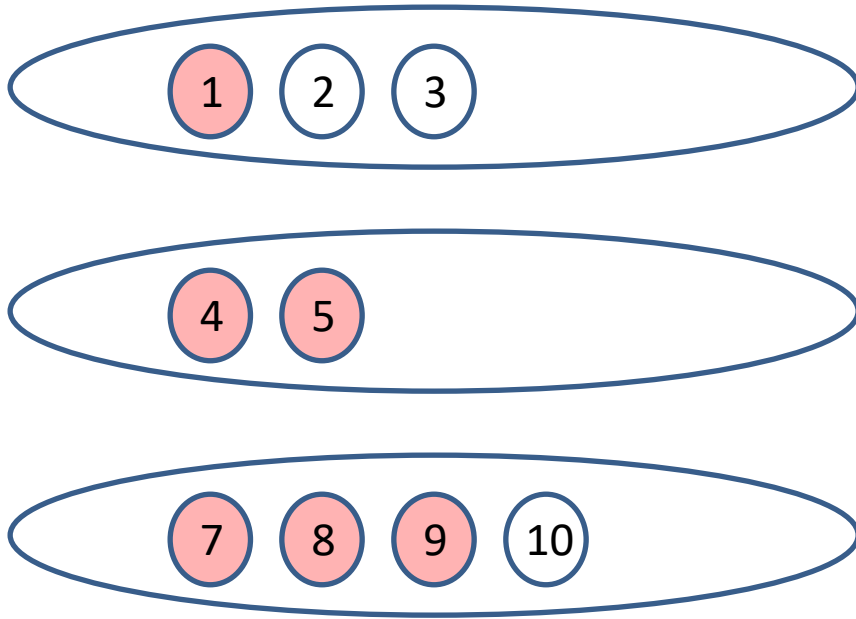
$$Q(S_1 \cup S_2 \cup S_4 - \{7, 8\})$$

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

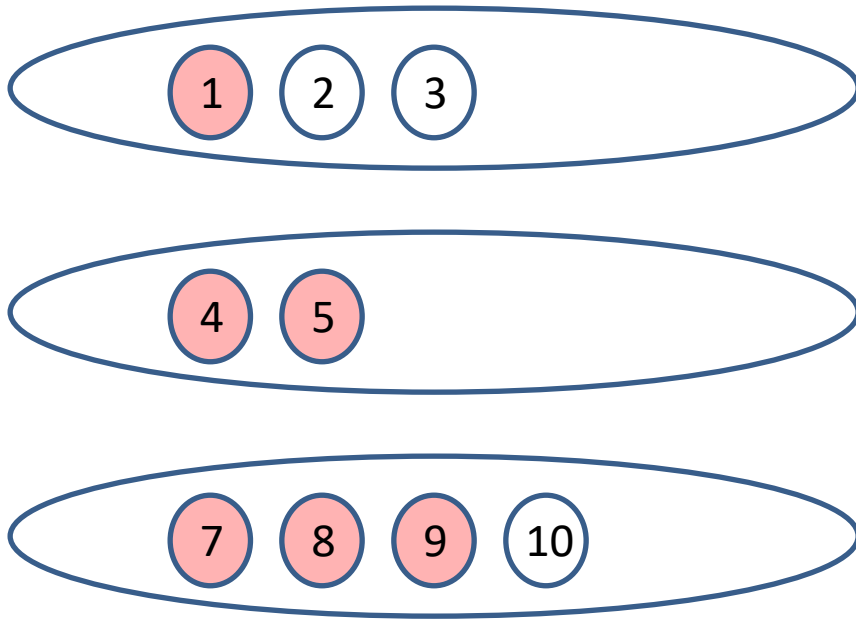
$$Q(S_1 \cup S_2 \cup S_4 - \{7,8,9\})$$

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)

Signals

- We live in a *digital world*.
- Telecommunication, entertainment, medical devices, gadgets, business – all revolve around digital media.
- Miniature sophisticated block-boxes process streams of bits accurately at high speeds.
- The digital world plays a fundamental role in our everyday routine, to such a point that we almost forget that we can not “hear” or “watch” these *streams of bits*, running behind the scenes.

Signal Processing

- A common goal of the engineering field of signal processing is to reconstruct a signal from **a series of sampling measurements**.
- In general, this task is impossible because there is no way to reconstruct a signal during the times that the signal is not measured. Nevertheless, **with prior knowledge or assumptions about the signal**, it turns out to be possible to perfectly reconstruct a signal from a series of measurements. Over time, engineers have improved their understanding of which assumptions are practical and how they can be generalized.



Sampling

- In signal processing, **sampling** is the reduction of a continuous-time signal to a discrete-time signal. A common example is the conversion of a sound wave (a continuous signal) to a sequence of samples (a discrete-time signal).
- A **sample** is a value or set of values at a point in time and/or space.
- A **sampler** is a subsystem or operation that extracts samples from a continuous signal.
- A theoretical **ideal sampler** produces samples equivalent to the instantaneous value of the continuous signal at the desired points.

Sampling idea

- Sampling can be done for functions varying in space, time, or any other dimension, and similar results are obtained in two or more dimensions.
- For functions that vary with time, let $s(t)$ be a continuous function (or "signal") to be sampled, and let sampling be performed by measuring the value of the continuous function every T seconds, which is called the **sampling interval** or the **sampling period**.^[1] Then the sampled function is given by the sequence: $s(nT)$, for integer values of n .

Key points

- Reconstructing a continuous function from samples is done by **interpolation algorithms**. The Whittaker–Shannon interpolation formula is mathematically equivalent to an ideal lowpass filter whose input is a sequence of Dirac delta functions that are modulated (multiplied) by the sample values. When the time interval between adjacent samples is a constant (T), the sequence of delta functions is called a Dirac comb. Mathematically, the modulated Dirac comb is equivalent to the product of the comb function with $s(t)$. That purely mathematical abstraction is sometimes referred to as *impulse sampling*.

Compressed Sensing

- **Compressed sensing** is a signal processing technique for efficiently acquiring and reconstructing a signal, by **finding solutions to underdetermined linear systems**.
- This is based on the principle that, through optimization, the sparsity of a signal can be exploited to recover it from far fewer samples than required by the Shannon-Nyquist sampling theorem.
- There are two conditions under which recovery is possible. The first one is sparsity which requires the signal to be sparse in some domain. The second one is incoherence which is applied through the isometric property which is sufficient for sparse signals.

Break Through

- An early breakthrough in signal processing was the [Nyquist–Shannon sampling theorem](#). It states that if the signal's highest frequency is less than half of the sampling rate, then the signal can be reconstructed perfectly by means of [sinc interpolation](#).
- The main idea is that with prior knowledge about constraints on the signal's frequencies, fewer samples are needed to reconstruct the signal.
- Around 2004, [Emmanuel Candès](#), [Terence Tao](#), and [David Donoho](#) proved that given knowledge about a signal's [sparsity](#), the signal may be reconstructed with **even fewer samples than the sampling theorem requires**.

Principle

- An underdetermined system of linear equations has more unknowns than equations and generally **has an infinite number of solutions**.
- In order to choose a solution to such a system, one must impose **extra constraints or conditions** (such as smoothness) as appropriate.
- In compressed sensing, one adds the constraint of **sparsity**, allowing only solutions which have a small number of nonzero coefficients. **Not all underdetermined systems of linear equations have a sparse solution.** *However, if there is a unique sparse solution to the underdetermined system, then the compressed sensing framework allows the recovery of that solution.*

Important facts

- Compressed sensing takes advantage of the redundancy in many interesting signals—they are not pure noise. In particular, many signals are sparse, that is, they contain many coefficients close to or equal to zero, when represented in some domain.
- Compressed sensing typically starts with taking a weighted linear combination of samples also called compressive measurements in a basis different from the basis in which the signal is known to be sparse. The results found by Emmanuel Candès, Justin Romberg, Terence Tao and David Donoho, showed that the number of these compressive measurements can be small and still contain nearly all the useful information.

Total Variation

- Total variation can be seen as a non-negative real-valued functional defined on the space of real-valued functions (for the case of functions of one variable) or on the space of integrable functions (for the case of functions of several variables).
- For signals, especially, total variation refers to the integral of the absolute gradient of the signal. In signal and image reconstruction, it is applied as total variation regularization where the underlying principle is that signals with excessive details have high total variation and that removing these details, while retaining important information such as edges, would **reduce the total variation of the signal and make the signal subject closer to the original signal in the problem.**

Another approach

- Could we use **combinatorial designs** to recover the signals?
- In general, the answer is “**No**”.
- In fact, there is no way to guarantee the **perfect recovery of the original signal**.
- So, it is up to how accurate we can get from the **measurements** we set up beforehand.
- **Discrete model is of course possible!**

Group Testing?

- In compressed sensing (*CS*), we are given an n -dimensional sparse signal with support size K .
- *A few* projections (*measurements*) of the sparse signal are obtained. (If the measurement is not orthogonal to the signal, then the inner product of the two vectors is not equal to zero.)
- *The goal is to identify the “support set” (the nonzero coordinates) while minimizing the number of projections.*

Binary Forms

- We may assume the signal **x** is an n -dim. $(0,1)$ -vector with a small number of 1's (sparse).
- The measurements (or the projections) are also an n -dim. $(0,1)$ -vectors and in total there are **t** of them.
- The inner product of the signal and the measurement is **1** if their inner product is larger than 0 and **0** otherwise.

Machine Learning

- Machine learning is a field of computer science for a long while. But, due to the emerge of Data Science, it becomes a more general topic in different expertise especially **AI**. As we can expect, there are tons of applications in human life.
- One of the main part of machine learning is "learning by examples", where the task is to approximate some unknown function when given its value at a number of specfic points. Therefore, this function learning
- problem can be tackled by a group testing approach, namely: **Non-adaptive Group Testing**.

Set up

- In a simple version, we have an unknown function $f : (Z_2)^n \rightarrow Z_2$ where $f(\mathbf{x}) = A\mathbf{x}$ such that for each row vector $\mathbf{a}$, the inner product of $\mathbf{a}$ and $\mathbf{x}$ is **1 if it is nonzero and 0 otherwise**.
- Here, we use **OR** for addition and **AND** for multiplication in binary set-up.
- We also assume that $\mathbf{a}$ contains at most d non-zero entries where d is very small comparing to n .
- The aim is to construct f (respectively A) using t measurements. Of course, we are looking for small t .
(Pooling designs!)

Constructions

- We may construct the measurement matrix by selecting its rows **randomly**.
- If the matrix has t rows, then we obtain an outcome vector which is a $t \times 1$ column vector.
- Noticed that the “0” components of the outcome vector **plays the most important role**.
- If the channel is **noise-free**, then we can use the **zero pool** to find a bunch of items which are not defective.

Signal: 010000010000

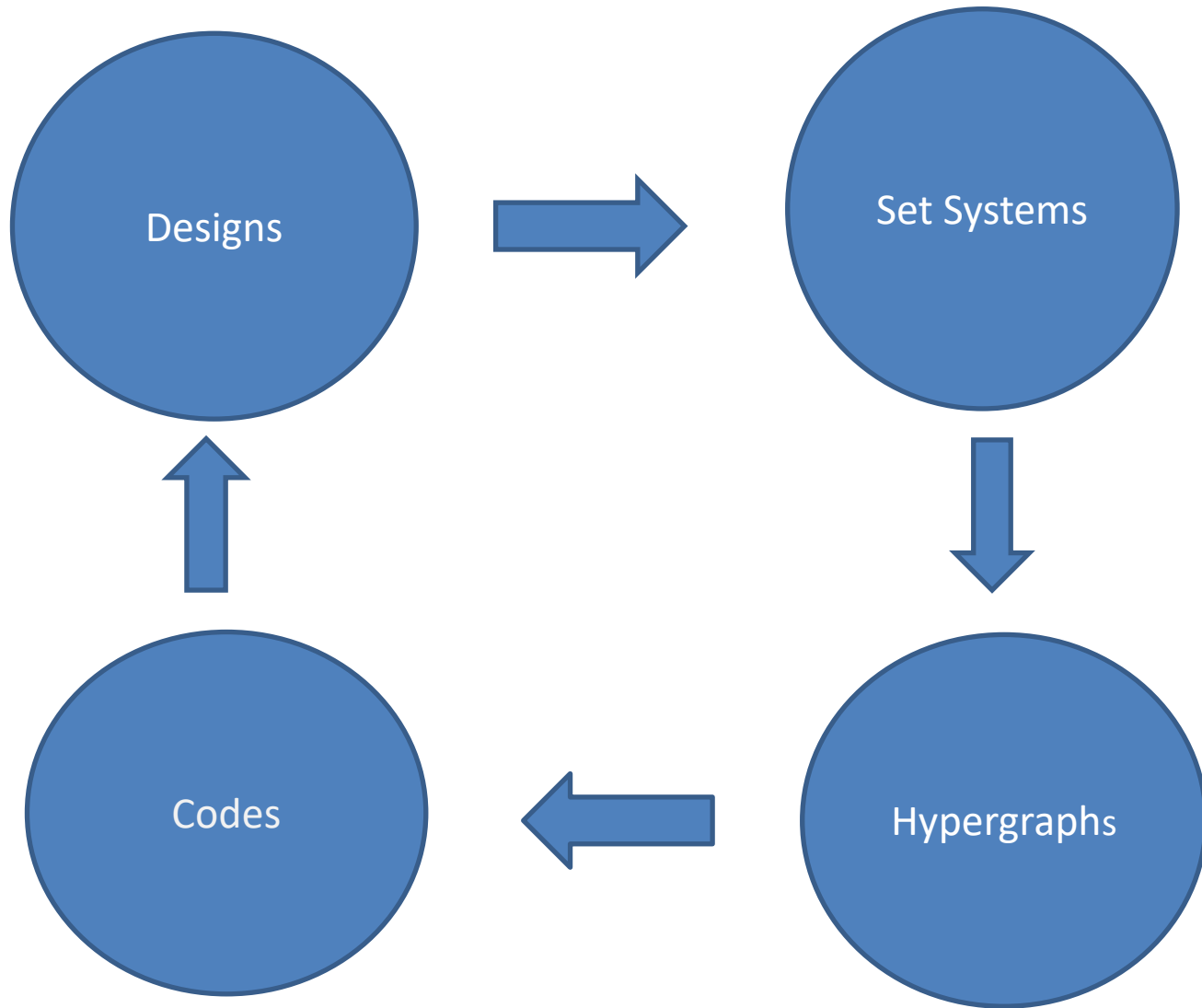
Outcome vector: 010111100

1			1			1			1		
1				1			1			1	
1					1			1			1
	1		1					1		1	
	1			1		1					1
	1				1		1		1		
		1	1				1				1
		1		1				1	1		
		1			1	1				1	

The blanks
are zeros.

Combinatorial Designs





Network Security

- **Network security** consists of the policies and practices adopted to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. **Users choose or are assigned an ID and password or other authenticating information that allows them access to information and programs within their authority.** Network security covers a variety of computer networks, both public and private, that are used in everyday jobs; conducting transactions and communications among businesses, government agencies and individuals.

Types of Networks

- Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions. It does as its title explains: **It secures the network, as well as protecting and overseeing operations being done.** The most common and simple way of protecting a network resource is by assigning it a unique name and a corresponding password.

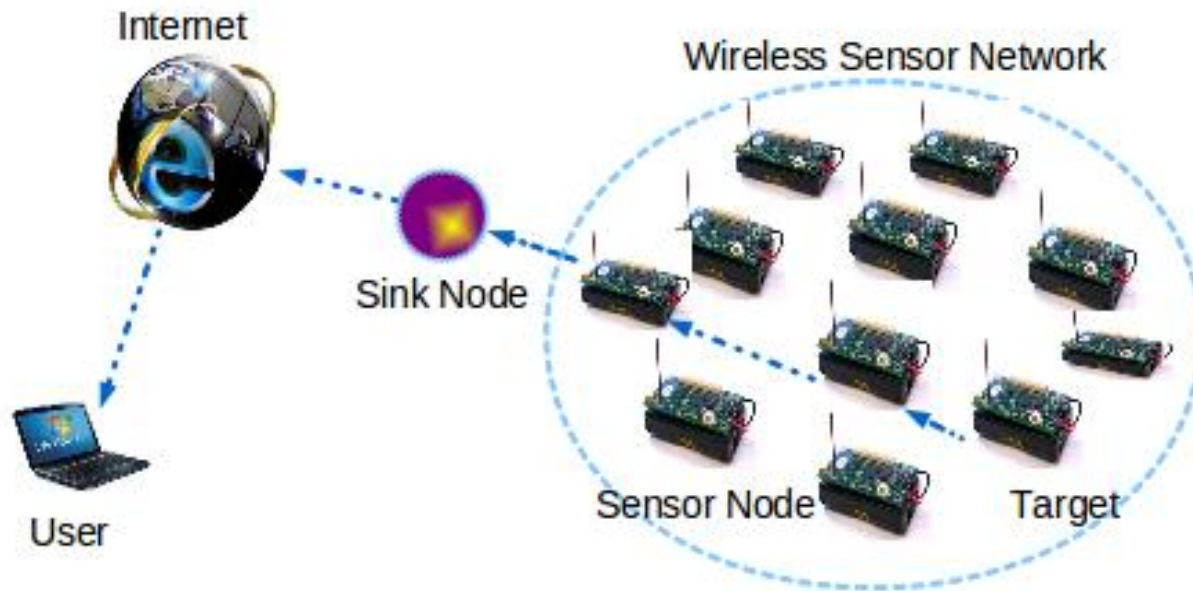
Wireless Sensor Networks

- **Wireless sensor networks (WSN)**, are similar to wireless ad hoc networks in the sense that they **rely on wireless connectivity and spontaneous formation of networks so that sensor data can be transported wirelessly**. Sometimes they are called **dust networks**, referring to minute sensors as small as dust. **Smart dust**^{[1][2][3]} is a U C Berkeley project sponsored by DARPA. Dust Networks Inc., is one of the early companies that produced wireless sensor network products. WSNs are spatially distributed autonomous sensors **to monitor physical or environmental conditions**, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main locations.

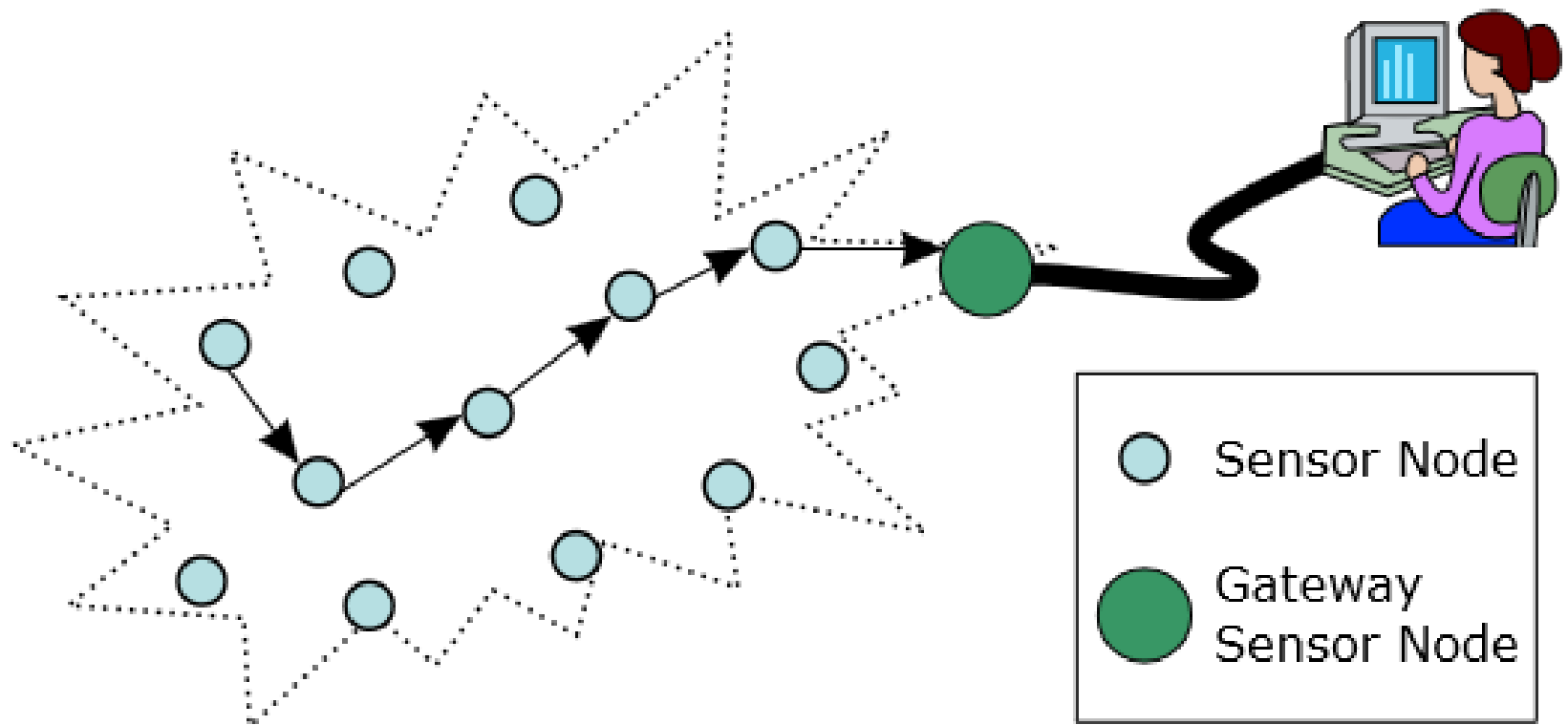
Useful Tools

- The more modern networks are **bi-directional**, also enabling *control* of sensor activity. The development of wireless sensor networks was motivated by **military applications such as battlefield surveillance**; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

Wireless Sensor Network



How it works?



Set Up

- The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a [radio transceiver](#) with an internal [antenna](#) or connection to an external antenna, a [microcontroller](#), an electronic circuit for interfacing with the sensors and an energy source, usually a [battery](#) or an [embedded](#) form of [energy harvesting](#). A [sensor node](#) might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created.

Cost of WSN

- The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.^{[4][5]}

Applications

- **Area monitoring**
- **Health care monitoring**
- **Environmental/Earth sensing**
- **Air pollution monitoring**
- **Forest fire detection**
- **Water quality monitoring**
- **Natural disaster prevention**
- **Industrial monitoring**

More on Data

- **Data center monitoring**

Due to the high density of servers racks in a data center, often cabling and IP addresses are an issue. To overcome that problem more and more racks are fitted out with wireless temperature sensors to monitor the intake and outtake temperatures of racks. As [ASHRAE](#) recommends up to 6 temperature sensors per rack, meshed wireless temperature technology gives an advantage compared to traditional cabled sensors.

- **Data logging**

Wireless sensor networks are also used for the collection of data for monitoring of environmental information, this can be as simple as the monitoring of the temperature in a fridge to the level of water in overflow tanks in nuclear power plants. The statistical information can then be used to show how systems have been working. The advantage of WSNs over conventional loggers is the "live" data feed that is possible

Security

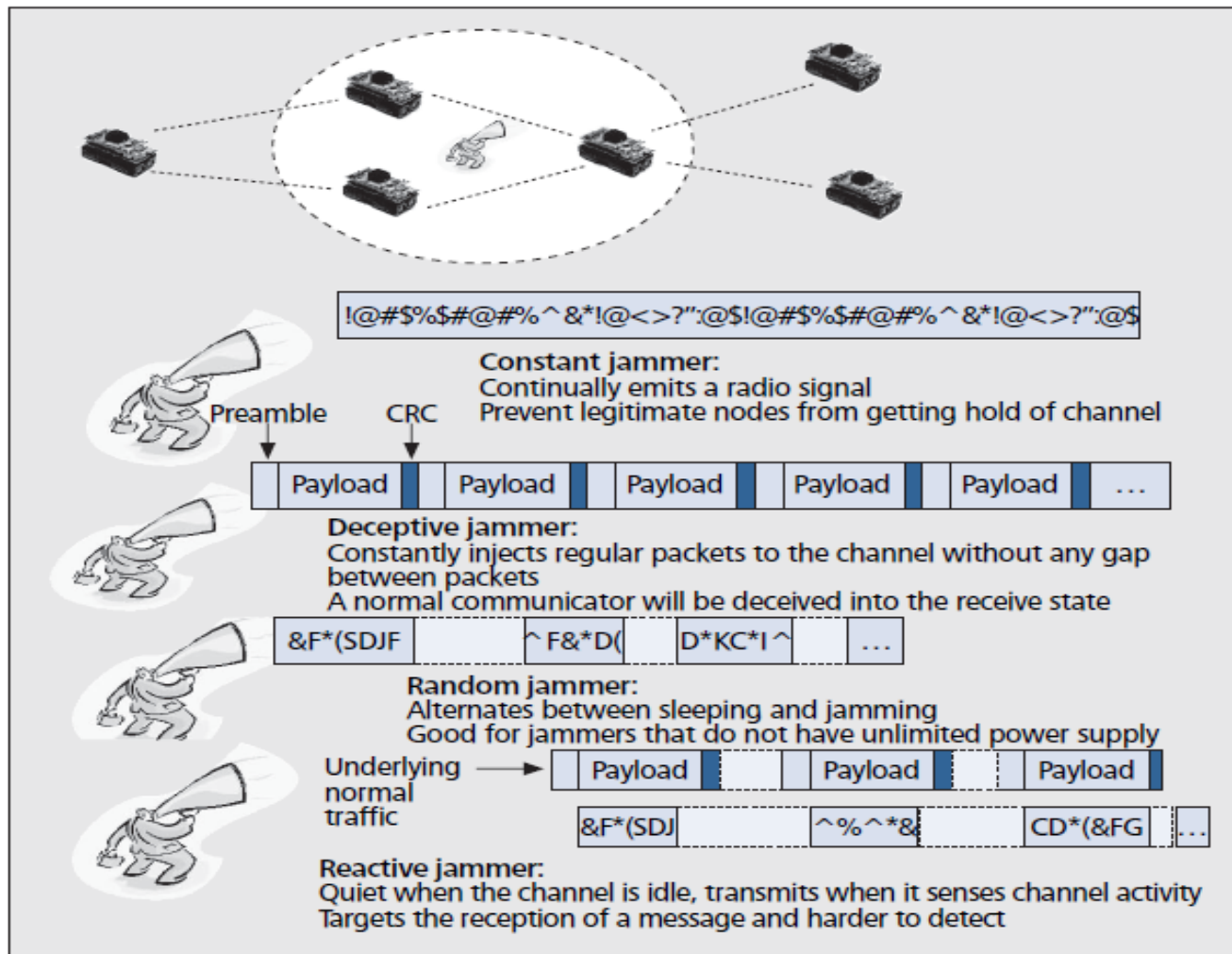
- Infrastructure-less architecture (i.e. no gateways are included, etc.) and inherent requirements (i.e. unattended working environment, etc.) of WSNs might pose several weak points that attract adversaries. **Therefore, security is a big concern when WSNs are deployed for special applications such as military and healthcare.** Owing to their unique characteristics, traditional security methods of computer networks would be useless (or less effective) for WSNs. Hence, lack of security mechanisms would cause intrusions towards those networks. These intrusions need to be detected and mitigation methods should be applied. More interested readers would refer to Butun *et al.*'s paper regarding intrusion detection systems devised for WSNs.

Types of Jammer

- Constant jammer
- Deceptive jammer
- Random jammer
- Reactive jammer



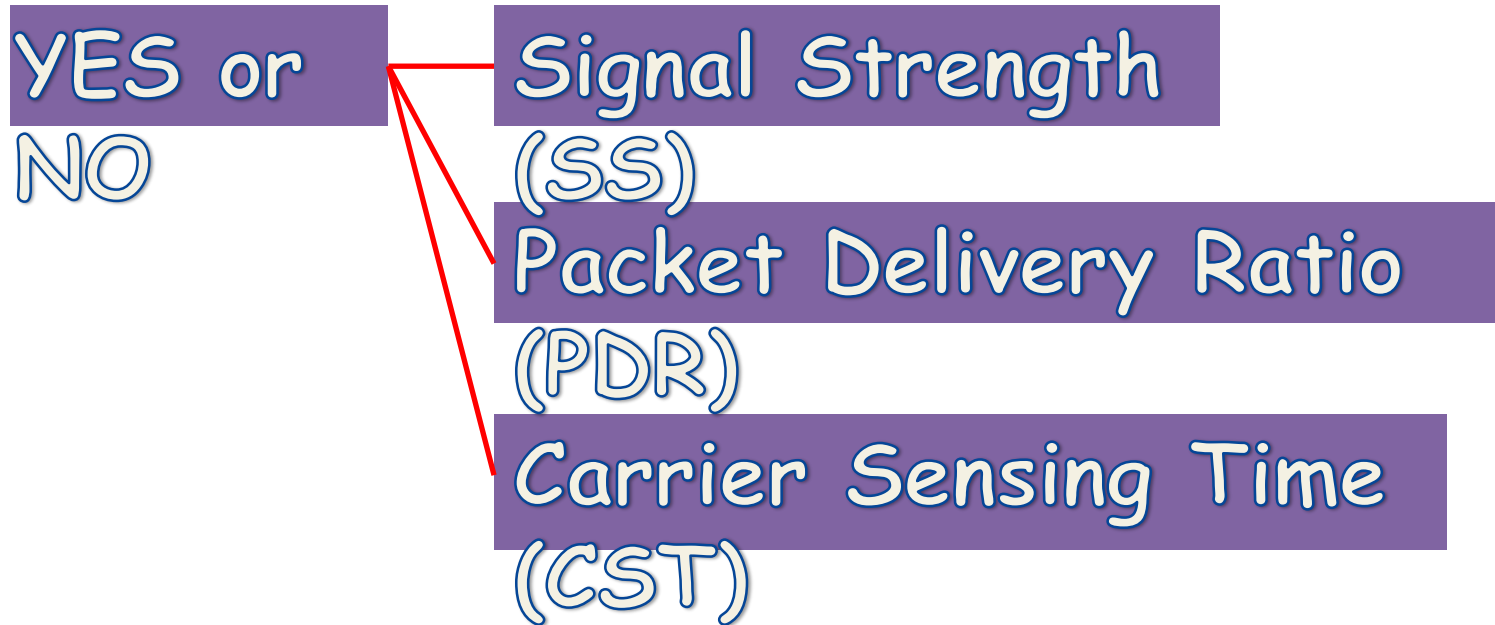
Jammer



Location of Jammers

- Finding the exact location of the jammer is very difficult task in wireless sensor network. The locations of jammers anywhere in wireless network and that disturbs the whole communication system. To overcome these challenges and increase the localization accuracy, we implement simulation that solves the jammer localization problem and get the high accuracy in our work. We display the result by bar graph it displays the comparison of existing system result with proposed systems result and actual result. Locations estimated are the true location of the jammer in the wireless sensor network, to improve our performance we adopt several algorithms that involve stochastic processes to approach the global optimum.

Detection for Jammers



Further Information

Coping with Reactive Jammers

Finding

Trigger nodes

1. Constructing a routing algorithm :
triggers be only receivers.
2. For triggers to send,
applying channel surfing becomes efficient.
3. Properly utilizing victim nodes (VN).

Basic Model

- Jamming range R .
- Sensor transmission range r_s .
- Triggering range $r = r_s$.
- Jammer distance $> R$.

Triggering Range

$$SNR = \frac{P_a}{P_n} > \theta \quad P_a = \frac{P_s}{r^\xi} \cdot Y \quad \Rightarrow \quad r < \left(\frac{P_s Y}{P_n \theta} \right)^{\frac{1}{\xi}}$$

SNR : Signal-Noise Ratio

P_s : power of the sensor signal

P_a : arrived signal power

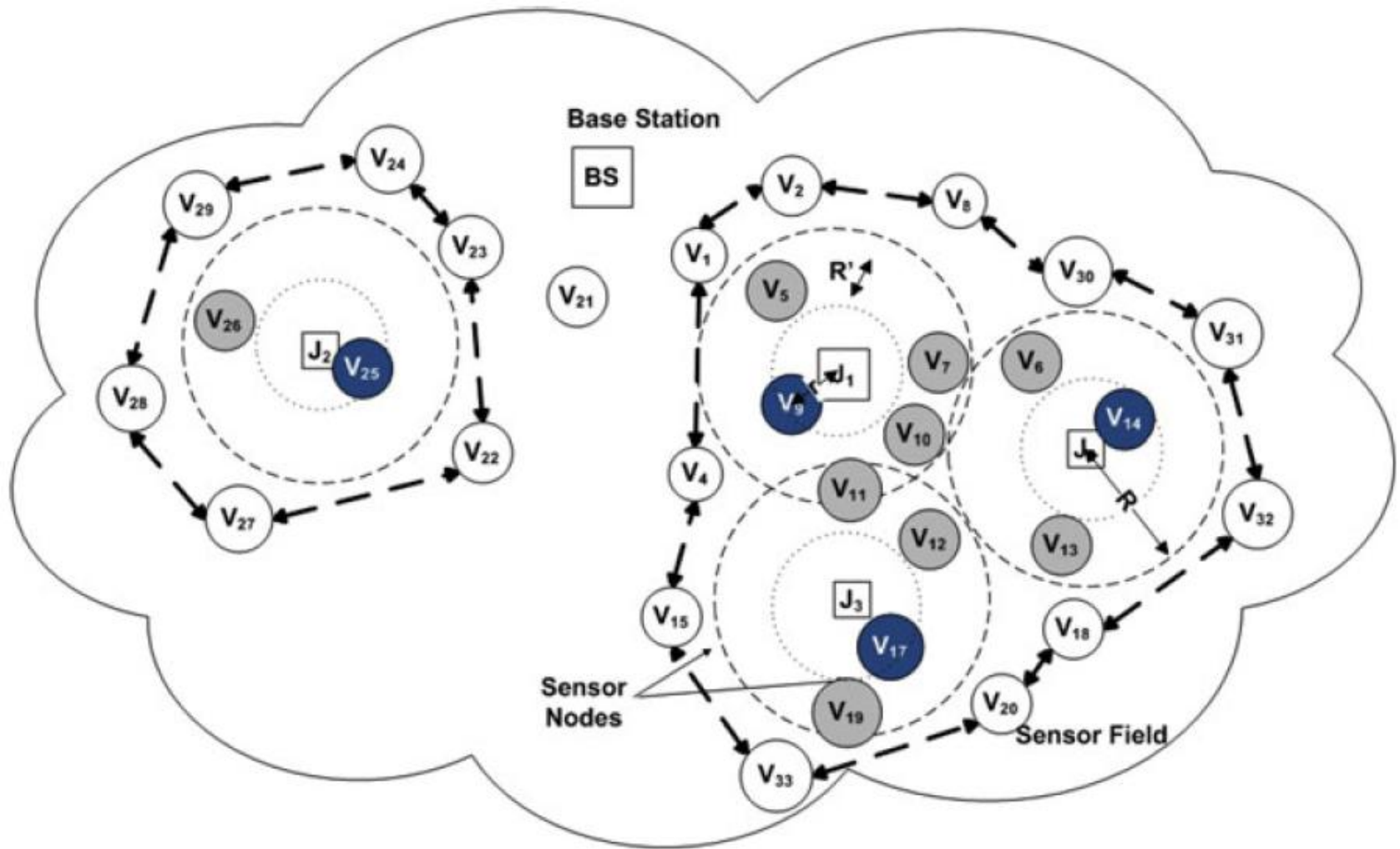
P_n : power of the background noise

θ : decision threshold

ξ : path – loss factor

Y : log - normally random variable

Basic Model



Identifying Trigger Nodes

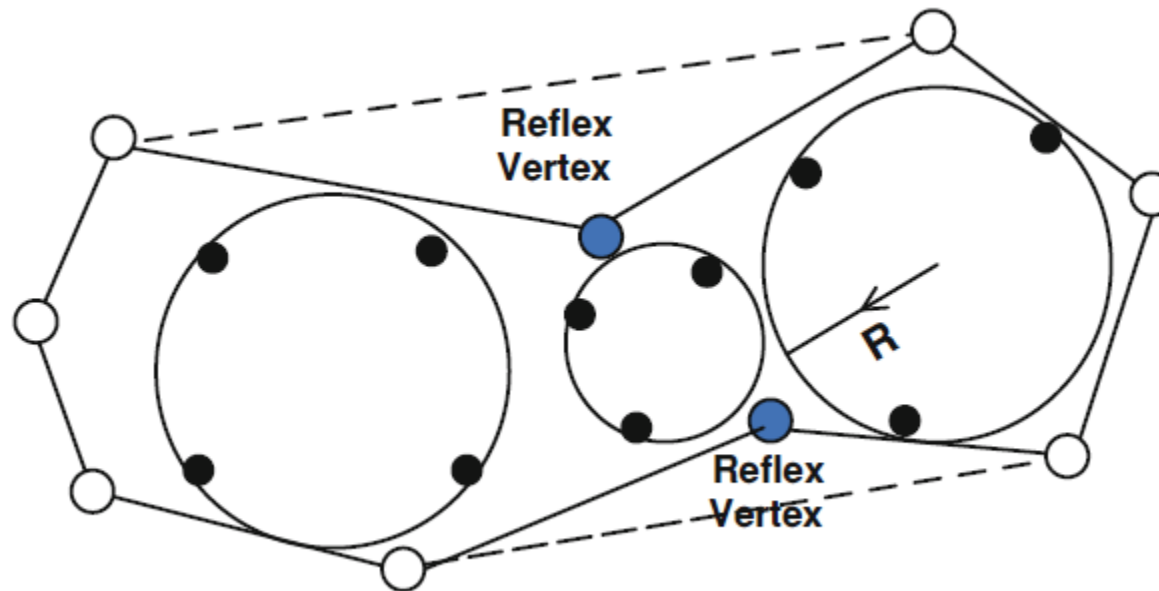
- Preprocessing

A diagram showing the 'Preprocessing' step branching into two sub-steps: 'Node Classification' and 'Jamming Range Estimation'. Red lines connect the word 'Preprocessing' to the start of each sub-step box.

 - Node Classification
 - Jamming Range Estimation
- Interference Free Group Testing (**IFGT**)
Algorithm
- Non-Adaptive Group Testing Detection (**NGTD**)
Algorithm

Jamming Range Estimation

- Dense-jammer case



- *Group testing?*
- *Please refer to the book
“Group Testing Theory in
Network Security” written by
My T. Thai.*

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), **Optimization Letters**, 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

- New bound on *2-bar-separable* codes of length 2 (with Minquan Cheng, J. Jiang, Yuan-Hsun Lo and Ying Miao), **Des. Code Cryptography**, to appear.
- Learning a *hidden graph* (with Huilan Chang and Chie-Huai Shih), under review of Optimization Letters.
- *Expanded separable codes* (with Minquan Cheng, J. Jiang, Yuan-Hsun Lo and Ying Miao), under review of IEEE Transaction of Information Theory.