

How to Search Efficiently?

Hung-Lin Fu 傅 恒 霖

Department of Applied Mathematics

The Importance of Search

- No matter who you are, you are living in searching for **something** almost every day.
- Sometimes **if you get lucky**, then you get what you look for.
- Sometimes, you **get nothing at all**. But, try again next time!



Outcomes

- 尋尋覓覓，冷冷清清，淒淒慘慘戚戚；
乍暖還寒時候，最難將息。-----李清照
- 眾裏尋他千百度，暮然回首；
那人卻在燈火闌珊處。-----辛棄疾
- Never give up! (永不放棄)

How to search?

- Mathematically we use an *algorithm*.

What is algorithm?

- A method that can be used by a computer for the solution of a problem.
- A sequence of computational steps that transform the input into the output.
- Examples of algorithms in the nature: DNA and cook book.
- The word "algorithm" comes from the name of [Abu Ja'far Mohammed ibn Musa al Khowarizmi](#) (780–825 A.D.), an Arab mathematician who wrote a textbook on mathematics.

Good or bad!

Why should we study algorithms?

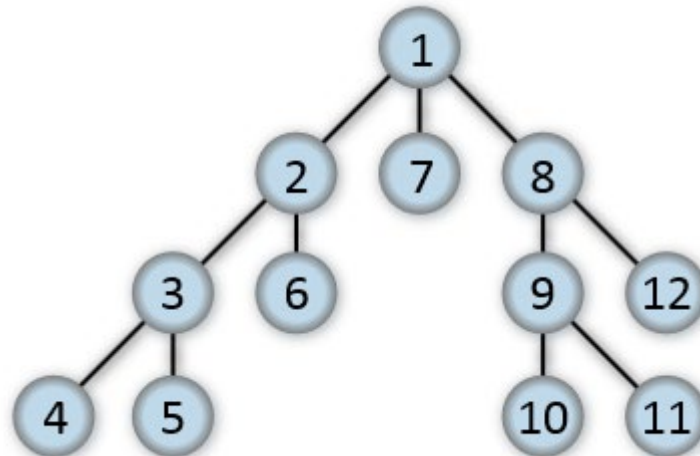
- A good algorithm implemented on a slow computer may perform much better than a bad algorithm implemented on a fast computer.

$f(n) \setminus n$	10	10^2	10^3
$\log_2 n$	3.3	6.6	10
n	10	10^2	10^3
$n \log_2 n$	0.33×10^2	0.7×10^3	10^4
n^2	10^2	10^4	10^6
2^n	1024	1.3×10^3	$> 10^{100}$
$n!$	3^6	$> 10^{100}$	$> 10^{100}$

Well-known Search Methods

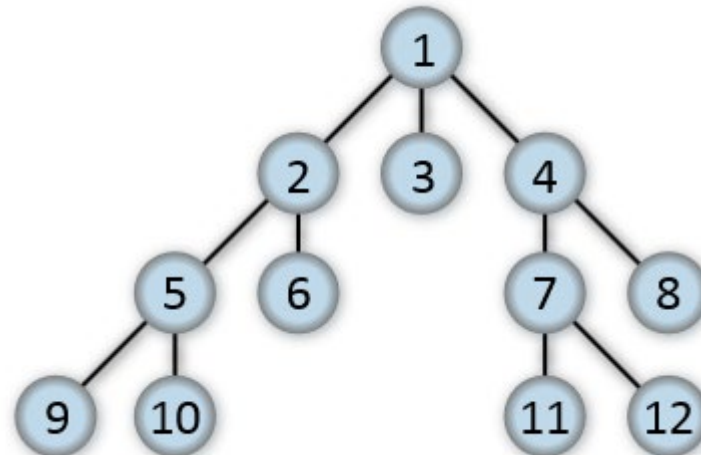
- **Depth-first search** (DFS)

One starts at the root (selecting some vertex as the root of a graph) and explores as far as possible along each branch before backtracking.



Breadth-first search

- It starts at the root (as above), sometimes referred as a **search key**, and explores the neighbor vertices first, before moving to the **next level neighbors** (distance 2 from the root and 3 and so on).



Ant Colony Algorithms

- The ant colony algorithm is an algorithm for finding optimal paths that is **based on the behavior of ants searching for food**.
- When an ant finds a source of food, it walks back to the colony leaving "markers" (pheromones) that show the path has food. When other ants come across the markers, they are likely to follow the path with **a certain probability**. If they do, they then populate the path with their own markers as they bring the food back. As more ants find the path, it gets stronger until there are a couple streams of ants traveling to various food sources near the colony.

Continued

- Because the ants drop pheromones every time they bring food, **shorter paths are more likely to be stronger**, hence optimizing the "solution." In the meantime, some ants are still randomly scouting for closer food sources. Once the food source is depleted, the route is no longer populated with pheromones and slowly decays.
- Because the ant-colony **works on a very dynamic system**, the ant colony algorithm works very well in graphs with changing topologies. Examples of such systems include computer networks, and artificial intelligence simulations of workers.

A fairly *new* way

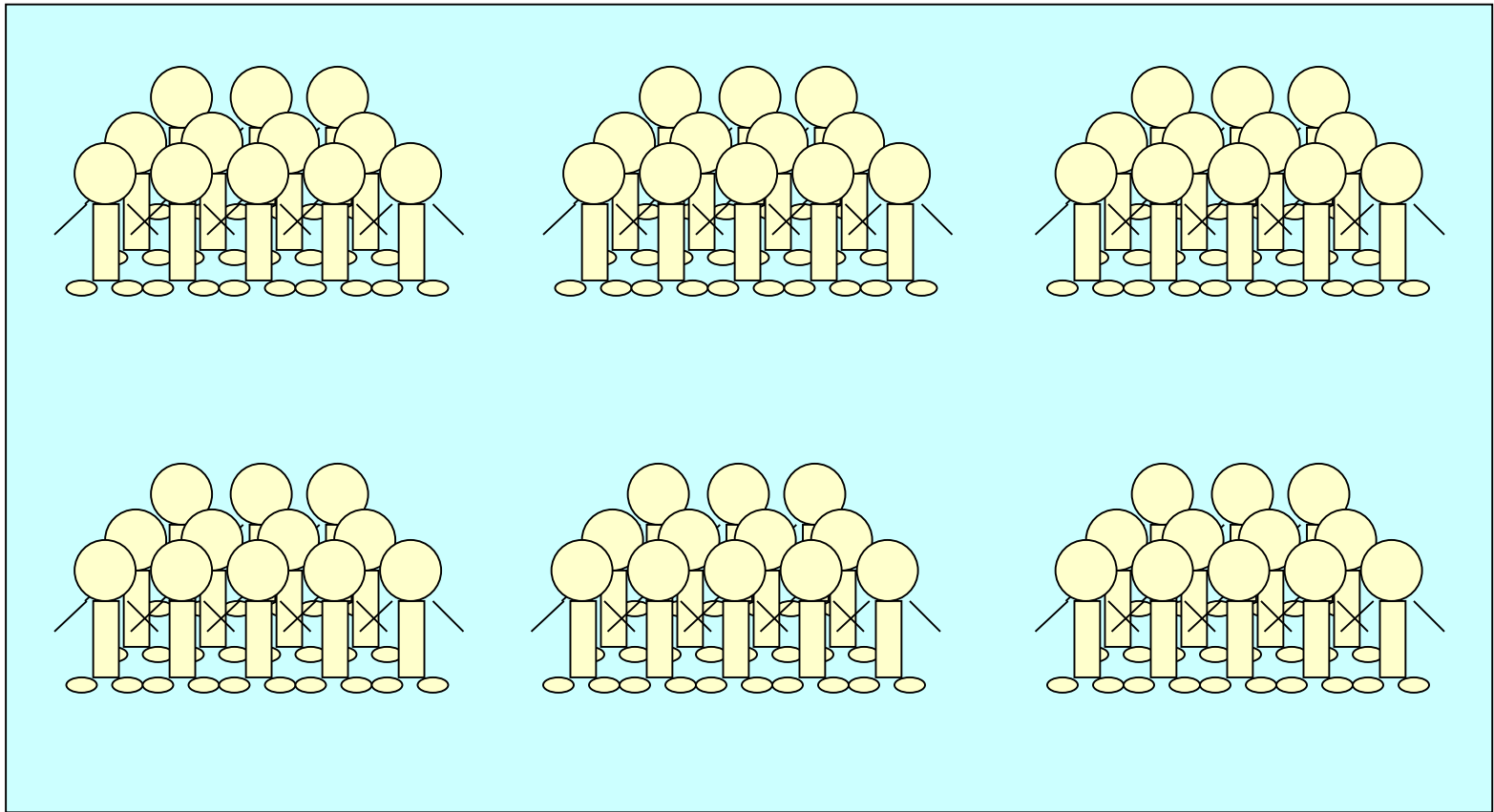
- The history says this idea was obtained at around 1943, but they in fact did not use it.
- The real impact comes at around year 2000, they applied the idea to **accelerate the process of DNA sequencing**.
- Now, it plays an important role in many topics which include Data Analysis.

Group Testing

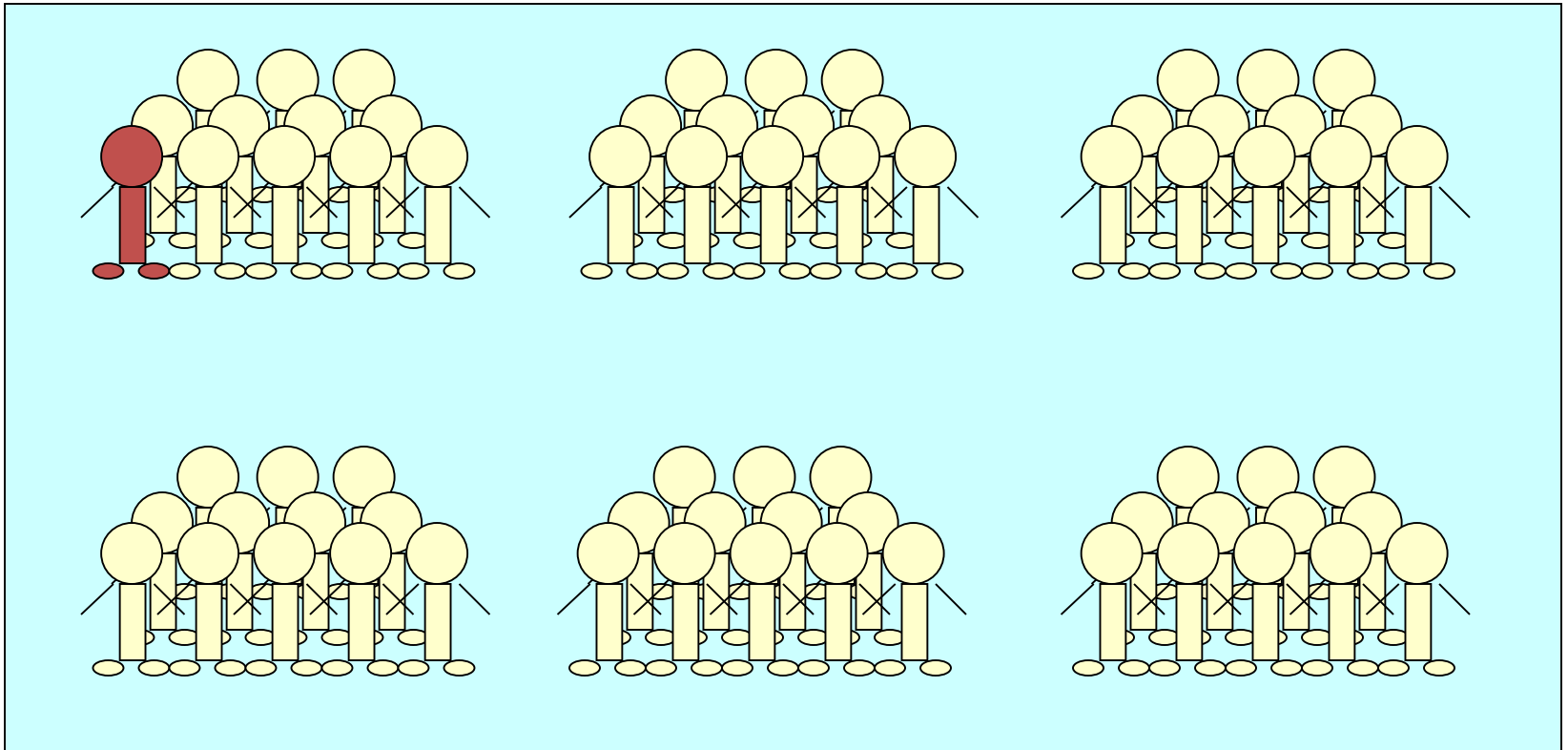
- Today, we shall mainly introduce this search methodology, namely *group testing*.
- This idea is very powerful in many applications not only on some special *algorithmic* topics.
- Despite of using computer, we also need a *smarter mind*!
- So, what is it? Let's start with an example.

Blood Testing

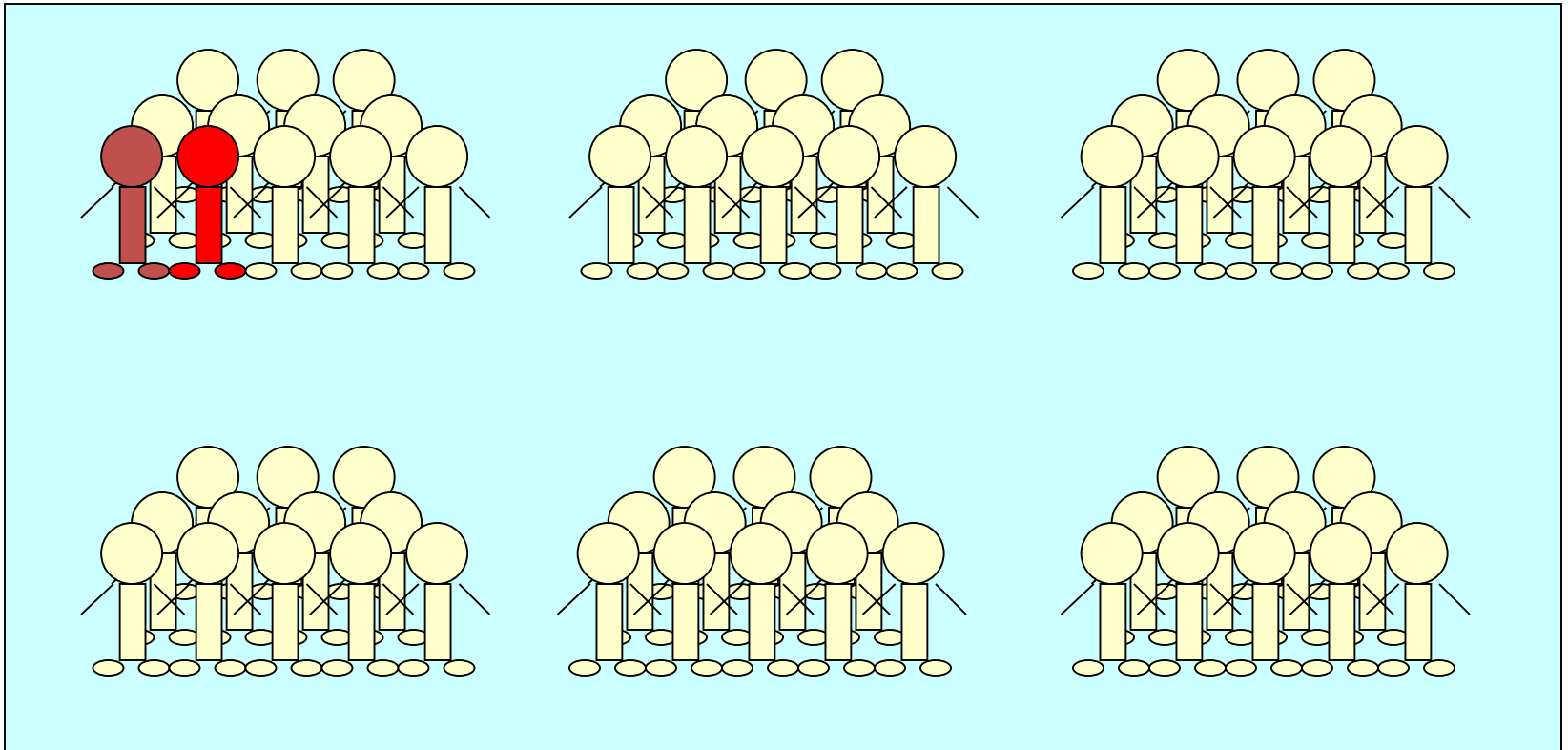
Syphilis (梅毒) Tests (World war II, 1943)



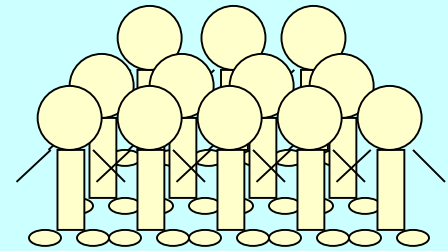
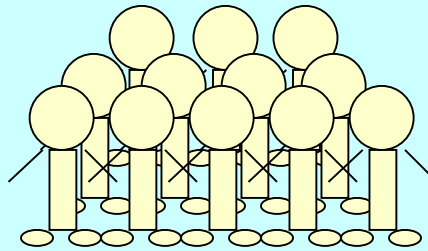
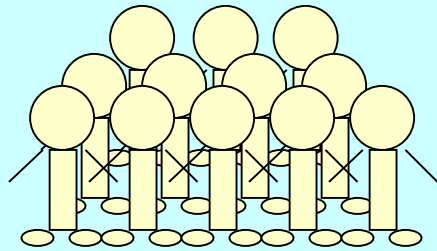
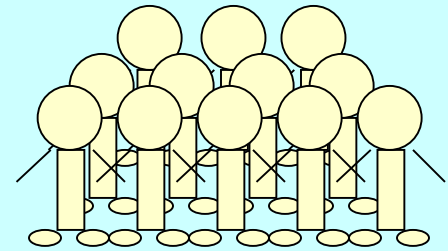
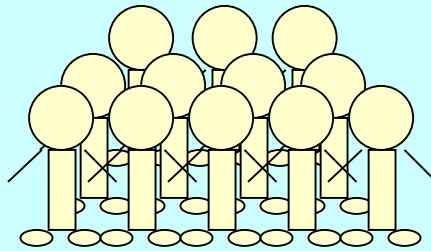
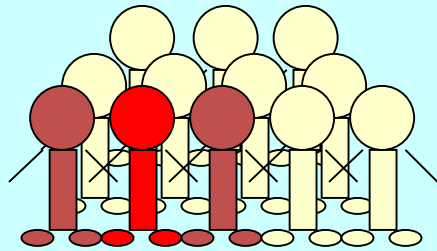
One by one



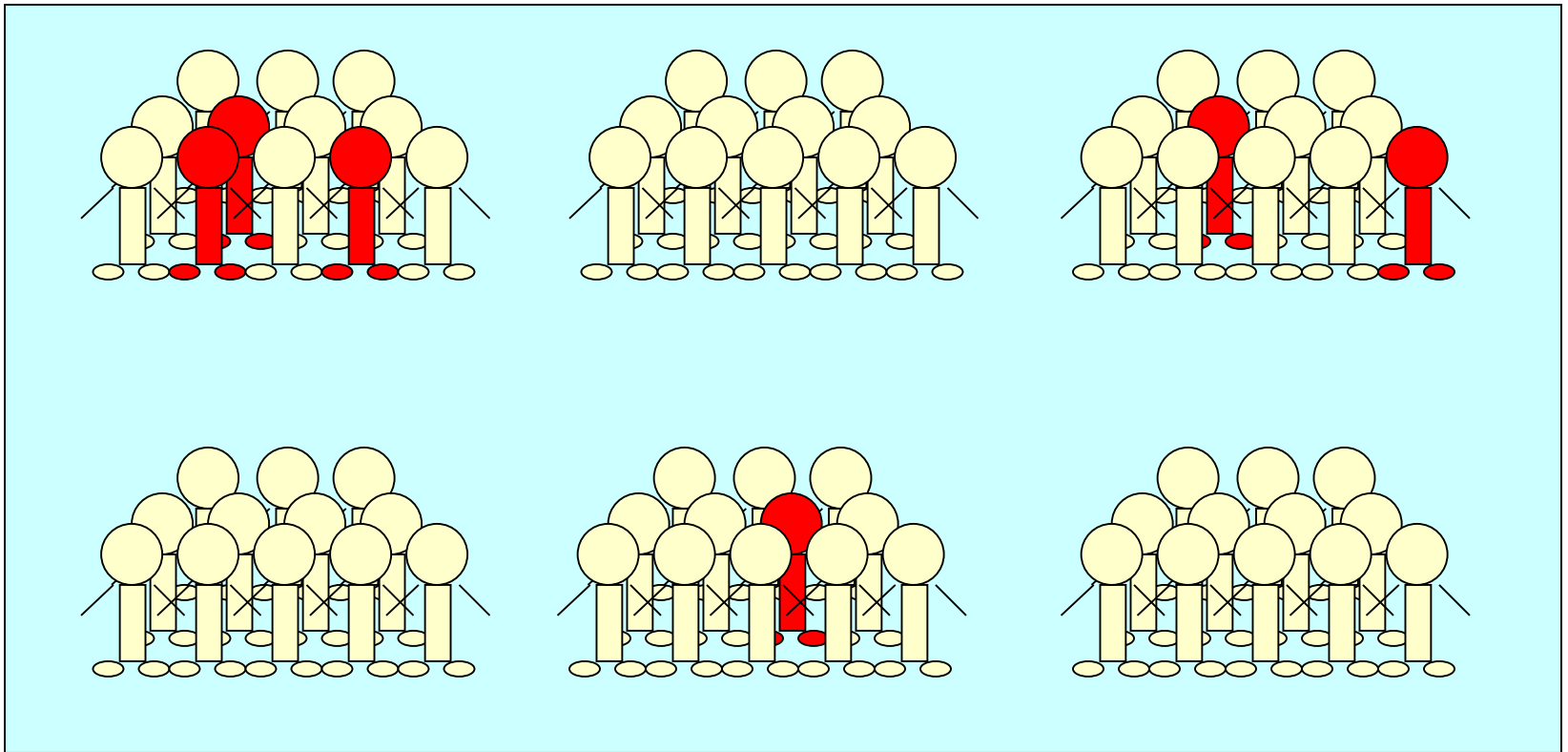
Red *is positive and* **Brown** *is negative*



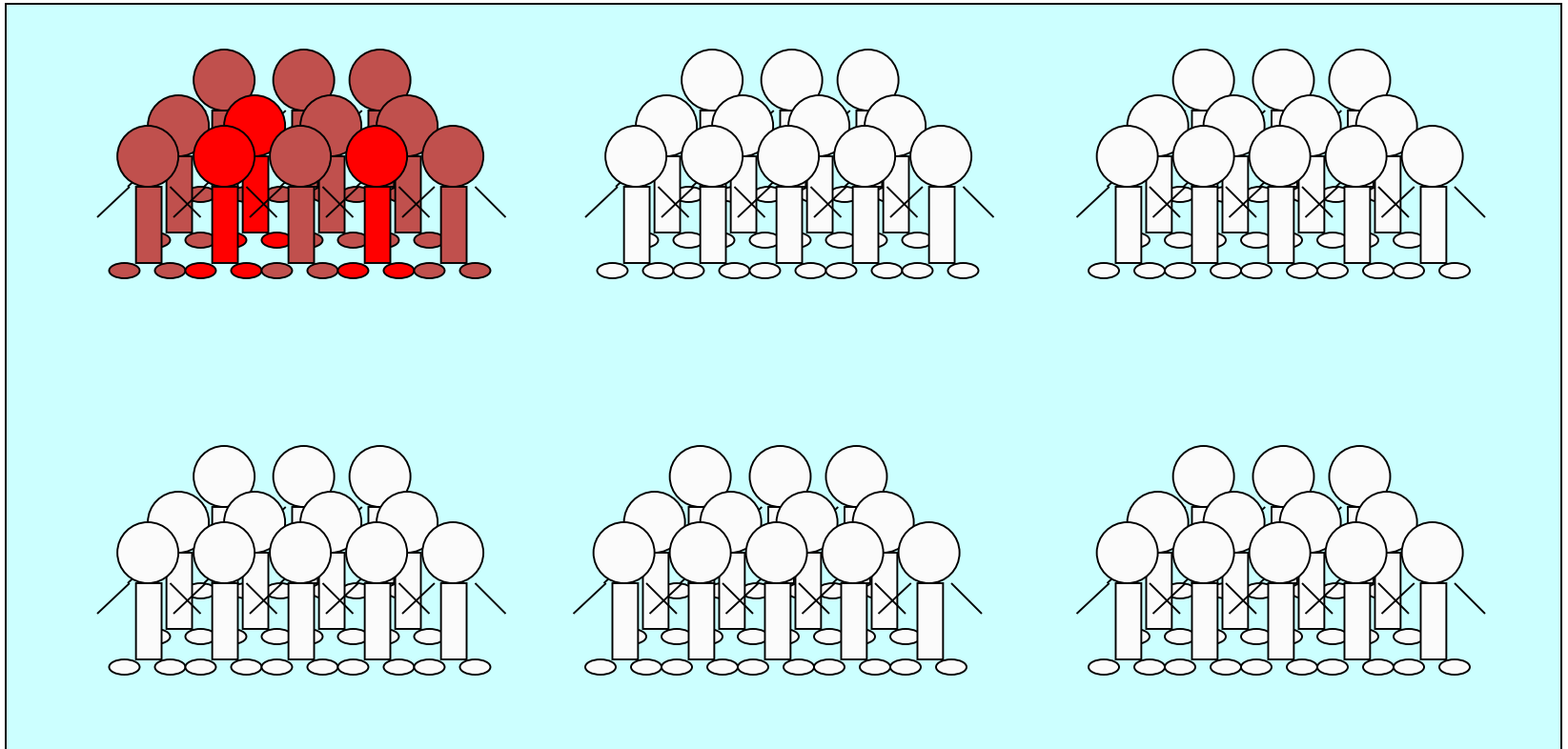
Keep going



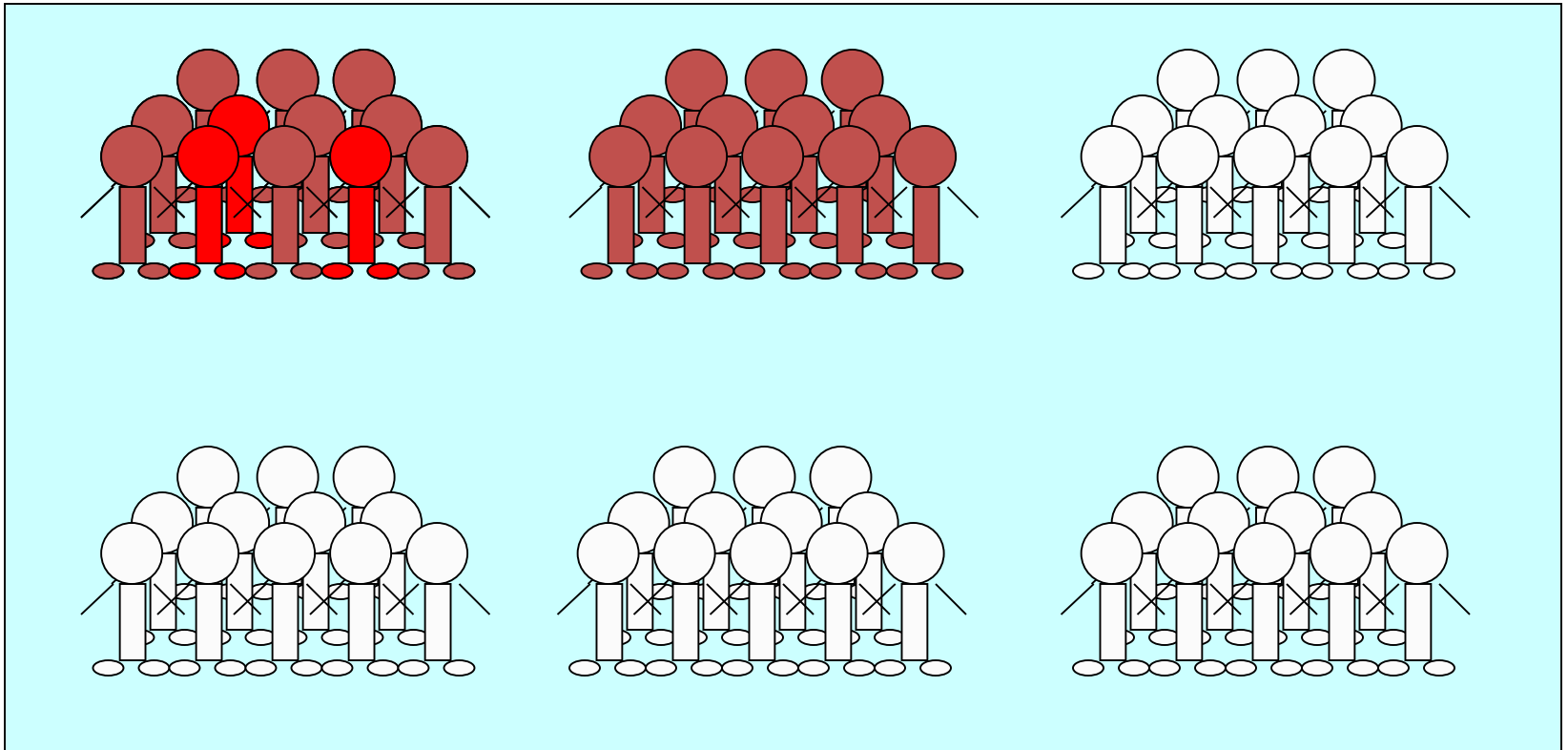
More positives (72 tests)



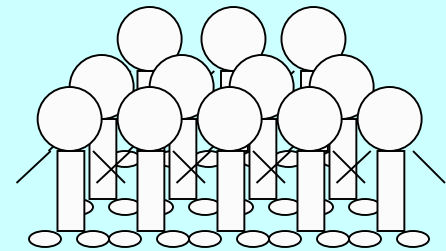
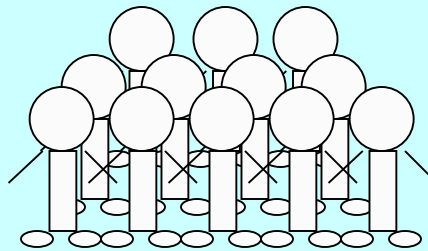
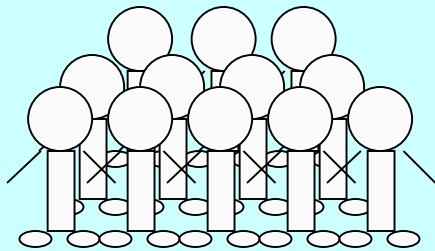
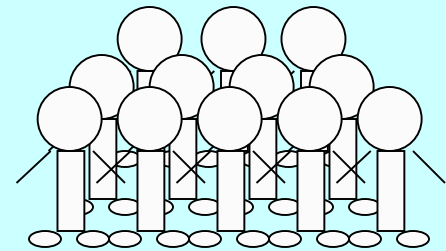
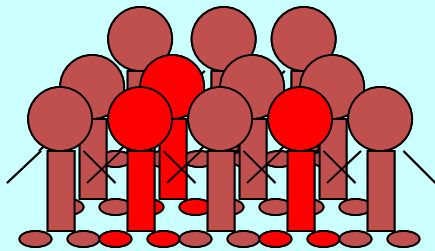
Test a group at one time



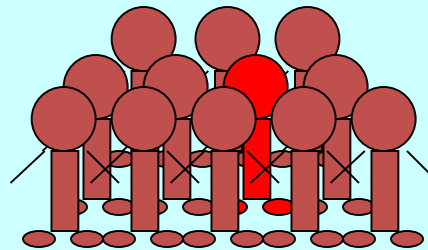
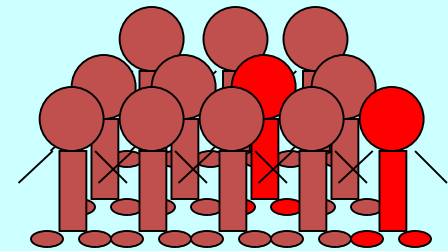
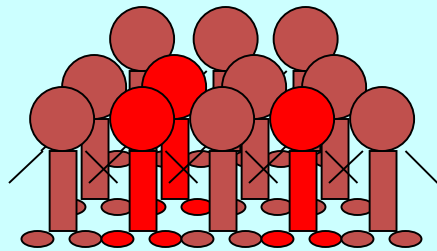
Some group is negative!



They are done!



Save Money! ?



Save 33-3 times

Another example to find a number

32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

2	3	6	7	10	11	14	15
18	19	22	23	26	27	30	31
34	35	38	39	42	43	46	47
50	51	54	55	58	59	62	63

4	5	6	7	12	13	14	15
20	21	22	23	28	29	30	31
36	37	38	39	44	45	46	47
52	53	54	55	60	61	62	63

8	9	10	11	12	13	14	15
24	25	26	27	28	29	30	31
40	41	42	43	44	45	46	47
56	57	58	59	60	61	62	63

1	3	5	7	9	11	13	15
17	19	21	23	25	27	29	31
33	35	37	39	41	43	45	47
49	51	53	55	57	59	61	63

16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Group testing (General Model)

- 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 one or how many.

Adaptive (Sequential) and Non-adaptive

- Can you see the **difference** between the above two ways in finding the answer?
- The first one (*adaptive or sequential*): 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.
- The second one (*non-adaptive*): You can ask all the questions (queries) *at the same time*.

Algorithms

- Adaptive algorithm
- Non-adaptive algorithm
- k-stage algorithm

The most popular one is a 2-stage algorithm in which we use one of the above two basic types of algorithm first and then in the second stage we test the left “suspected” items one by one.

Save Money or Time

- 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. (*Save money!?*)
- 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. (*Save time!*)

Non-adaptive Algorithm

We can use a matrix to describe a non-adaptive algorithm.

Items are indexed by columns and the tests are indexed by rows.

Therefore, the (i, j) entry is **1** if the item j is included in the *pool* i (for test), and **0** otherwise.

An example (12 items and 9 tests)

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.

Relation with Coding Theory

- We can look at the first column and relate it with the codeword **111000000**, and the second one **000111000**, etc.
- So we have a code with length **9** and size **12**.
- **Coding Theory** is very useful in *communication*.
- The code obtained above is known as a *binary code*.

Relation with designs

- Let $M = [m_{i,j}]$ be a $t \times n$ matrix mentioned above. Then we can use n sets (ordered) S_i 's to represent the matrix where

$$S_k = \{i : m_{i,k} = 1, i = 1, 2, \dots, t\}, k = 1, 2, \dots, n.$$

- The following sets represent a (0,1)-matrix:
 $\{1,2,3\}, \{4,5,6\}, \{7,8,9\}, \{1,4,7\}, \{2,5,8\}, \{3,6,9\}, \{1,5,9\},$
 $\{2,6,7\}, \{3,4,8\}, \{1,6,8\}, \{2,4,9\}, \{3,5,7\}.$ (Have you seen this collection of sets before?)
- It is known as an *Affine plane of order 3*.

Set notation

- It is easier to apply combinatorial structures to construct pooling designs, therefore, we use sets (or codewords) for columns in non-adaptive algorithms.
- We shall use set notation most of the time.
- The set corresponding to a codeword (binary vector) is called the *support* of the codeword.

Can we find positives from the above matrix?

- Yes, we can if the number of positives is not too many, say at most 2, by running the 9 tests simultaneously corresponding to rows.
- The reason is that the union of (at most) 2 columns can not contain any other distinct column. (?)

Decoding Idea

- Since there are 12 items and the number of positives is at most 2, we have $1 + 12 + 66$ possible inputs. (?)
- Let each input be a 12-dim column vector $\mathbf{x}$ and A be the above 9×12 matrix. Then the outcome $(0,1)$ -vector $\mathbf{y}$ is obtained by $A\mathbf{x}$ such that its i^{th} component is 1 if the i^{th} component of $A\mathbf{x}$ is positive and 0 otherwise.
- If A is 1-1, then we are able to find the positives, see it?

Outcome Vectors

- The vector **y** is an *outcome vector* which is corresponding to an input **x**.
- If A (*a linear transformation*) is 1-1, then we can decode **x** long as we know its outcome vector.
- In order to get the job done with lower decoding complexity, extra properties for A is needed.

d-separable and d-disjunct matrices

- A matrix is *d-separable* if $\cup D \neq \cup D'$ for any two distinct d-sets D and D' (columns), i.e. no two unions of d columns are the same.
- A matrix is *d-disjunct* if no column is contained in the union of any other d columns.
- A *d-disjunct* matrix is *d-separable*, *but the other way around may not be correct. (?)*

Important Facts

- *A d -disjunct matrix can be applied to find $k (\leq d)$ positives.*

Proof. The union of $k (\leq d)$ columns corresponds to distinct *outcome vector*. (This is also true for d -separable matrices.)

- *d -disjunct matrices have a simple decoding algorithm, namely, a column is positive if and only if it does not appear in a negative row.*

Various Models

- *Errors* occurred.
- There are *inhibitors*.
- There is a *threshold*.
- Defective items are sets: *complex* model.
- *Competitive* model: the number of defectives is unknown.
- The defectives are *mutually obscuring*.

Remark

- A *design* is a pair (X, B) where X is a v-set and B is a collection of subsets of X . So, the matrix mentioned above can be viewed as a *design*.
- A design defined on a v-set can be viewed as a *binary code of length v*.
- A design (X, B) can also be viewed as a *hypergraph* with vertex set X and edge set B .

Further Remarks

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

What if A is not a $(0,1)$ -matrix?

- We can convert A into a matrix with more rows by using binary representation.
- For example, if A is an m by n matrix defined on $S = \{0, 1, 2, \dots, q-1\}$ where $q = 2^k$, then we can find a matrix A^* to correspond to A where A^* has km rows and n columns. (See it?)
- Each element in S can be represented by a column vector with k components.

Another idea!

- We can deal with matrices defined on S mentioned above directly.
- The definitions of “separable” and “disjunct” will be different respectively from set inclusion.
- Here is an example:

0	1	1	3	2
1	3	1	2	0
1	0	2	2	1

Descendant set

- There are 5 codewords in the above example:
 c_1, c_2, c_3, c_4, c_5 .
- So, what is the descendant set of the first three codewords?
- *Guess???*
- $\{0,1\} \times \{1,3\} \times \{0,1,2\}$ is the answer.
- Why?

Human Genome Project



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*)。

Group Testing Works!

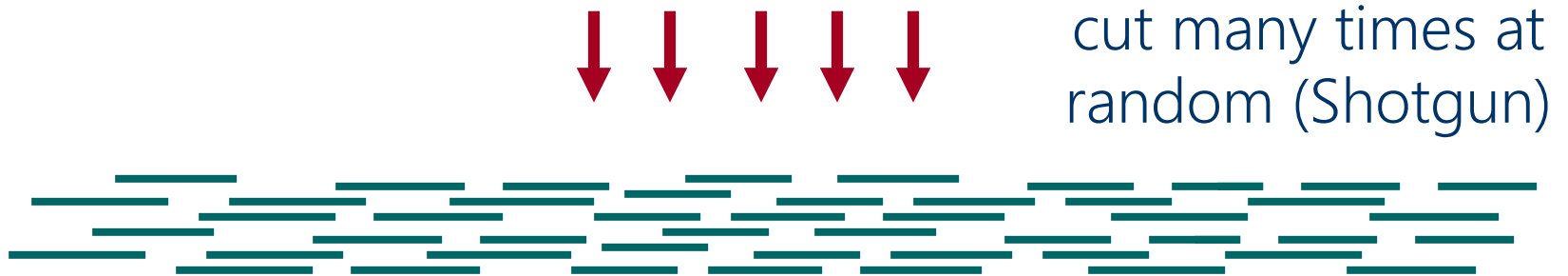
- In screening clone library the goal is to determine which clones in the clone library hybridize with a given probe in an efficient fashion. A clone is said to be **positive** if it hybridize with the **probe(探針)**, and **negative** otherwise.

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 ...**

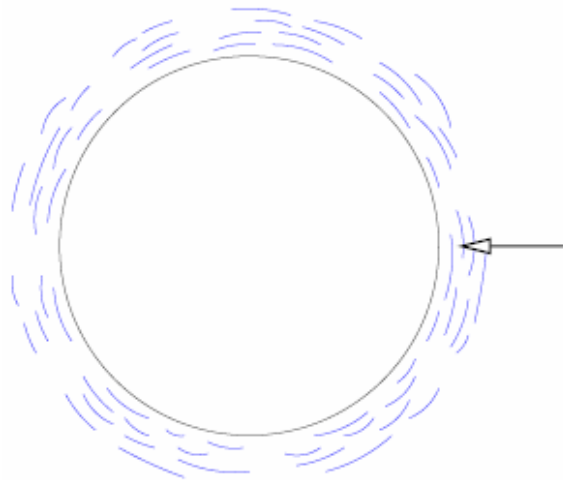
Random shotgun approach

genomic segment



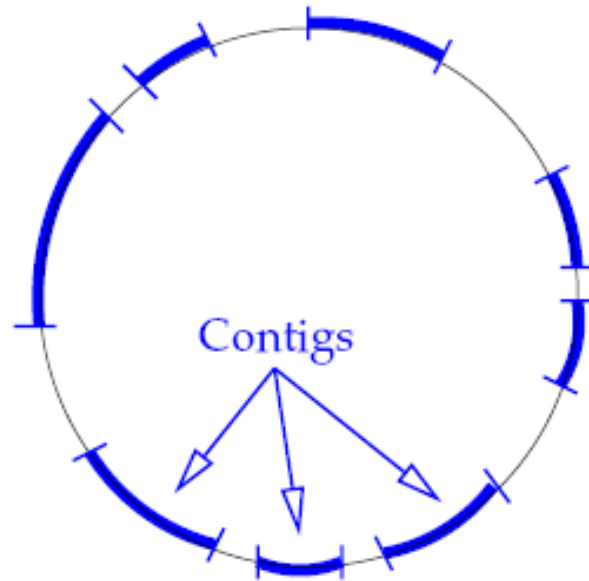
Whole-genome shotgun sequencing

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

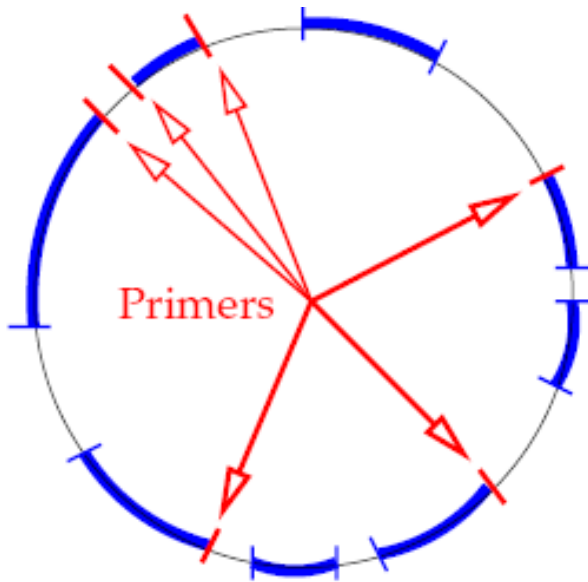


Circular genome

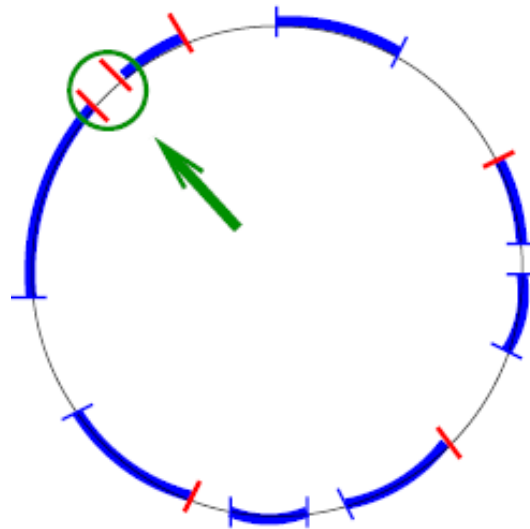
- Reads are assembled into contigs with unknown relative placement.

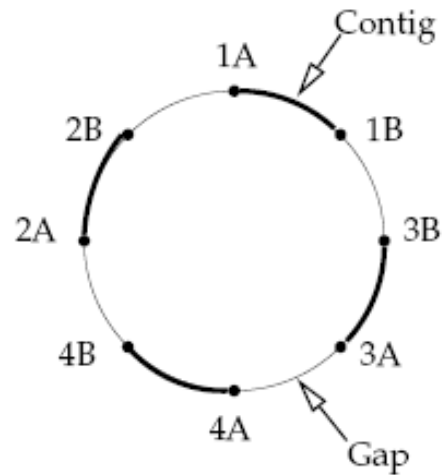


– **Primers** : (short) fragments of DNA characterizing ends of contigs.



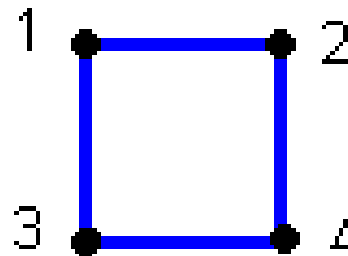
- A **PCR (Polymerase Chain Reaction)** reaction reveals if two primers are proximate (adjacent to the same gap).
- **Multiplex PCR** can treat multiple primers simultaneously and outputs if there is a pair of adjacent primers in the input set and even sometimes the number of such pairs.

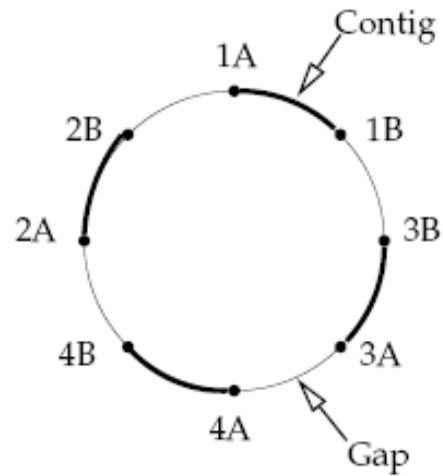




- Two primers of each contig are “mixed together”
 - Find a Hamiltonian cycle by PCRs!

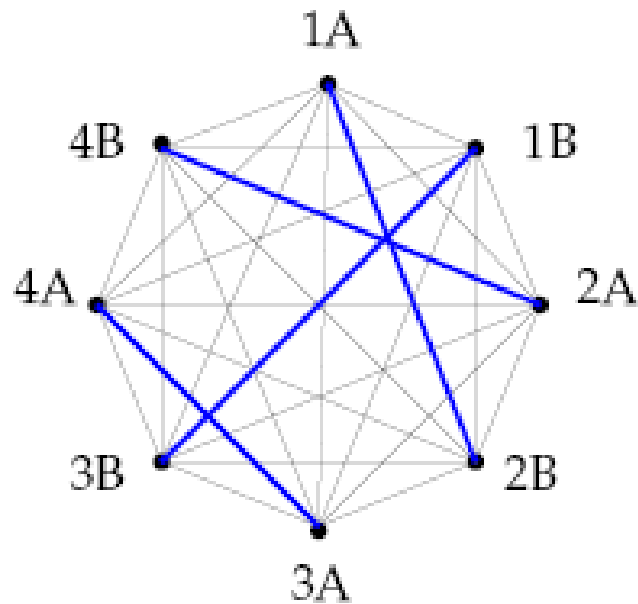
1A ——— 1B
 2A ——— 2B
 3A ——— 3B
 4A ——— 4B





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

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



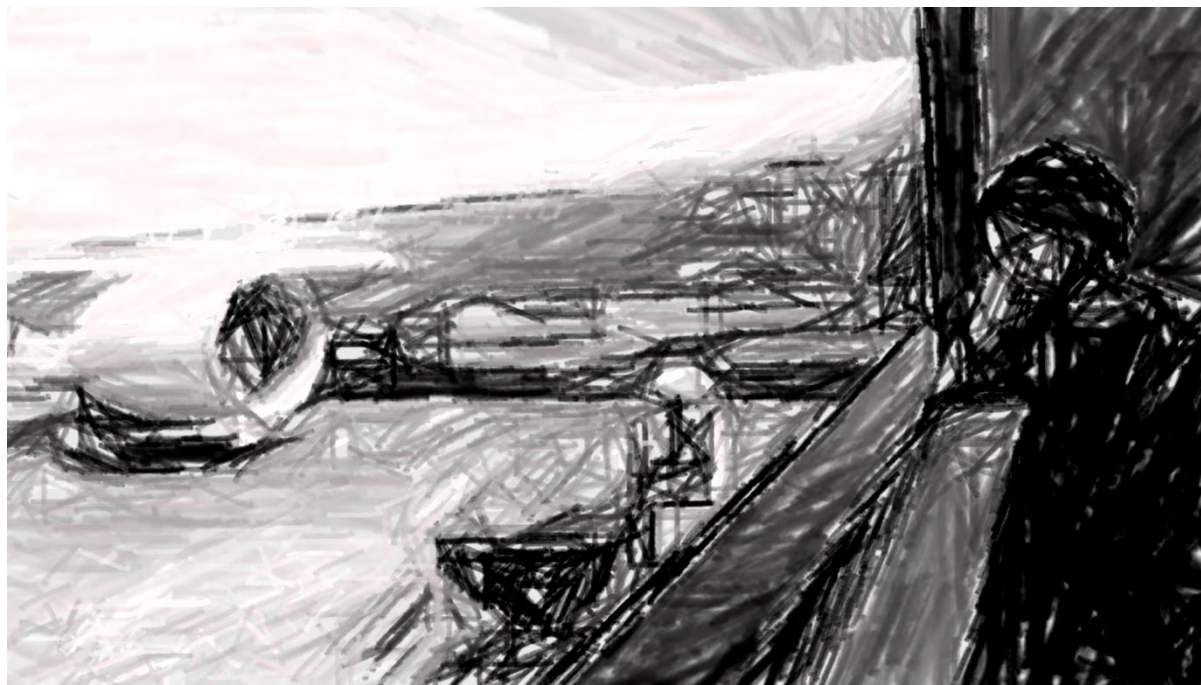
Goal

- 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.

Image processing: Another application

- There are ways of producing an image (**encoding**).
- But, transmitting the image takes more effort which includes how to **decode** the image sent.
- In general, we also consider the occurrence of *noise* during transmission.
- Sometimes, *fingerprinting* is needed and the idea of group testing shows its power.

• 搜 尋

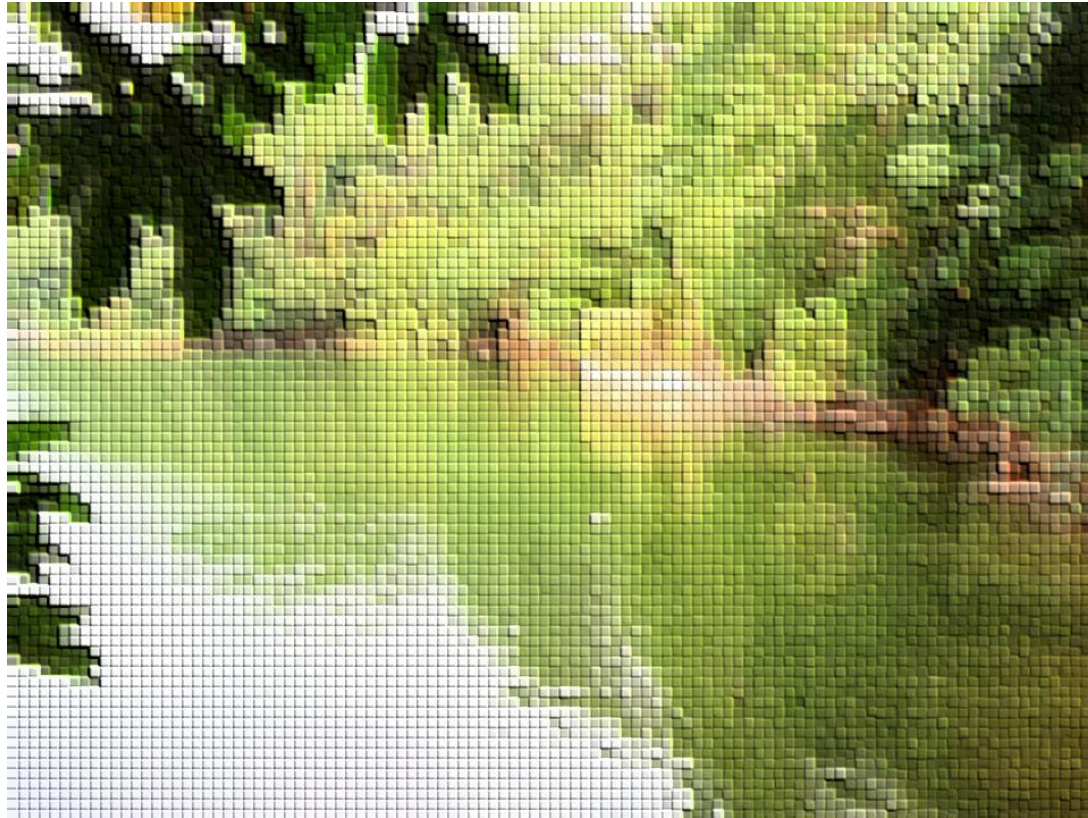




Not clear!



Digital Improvement!



Much better!



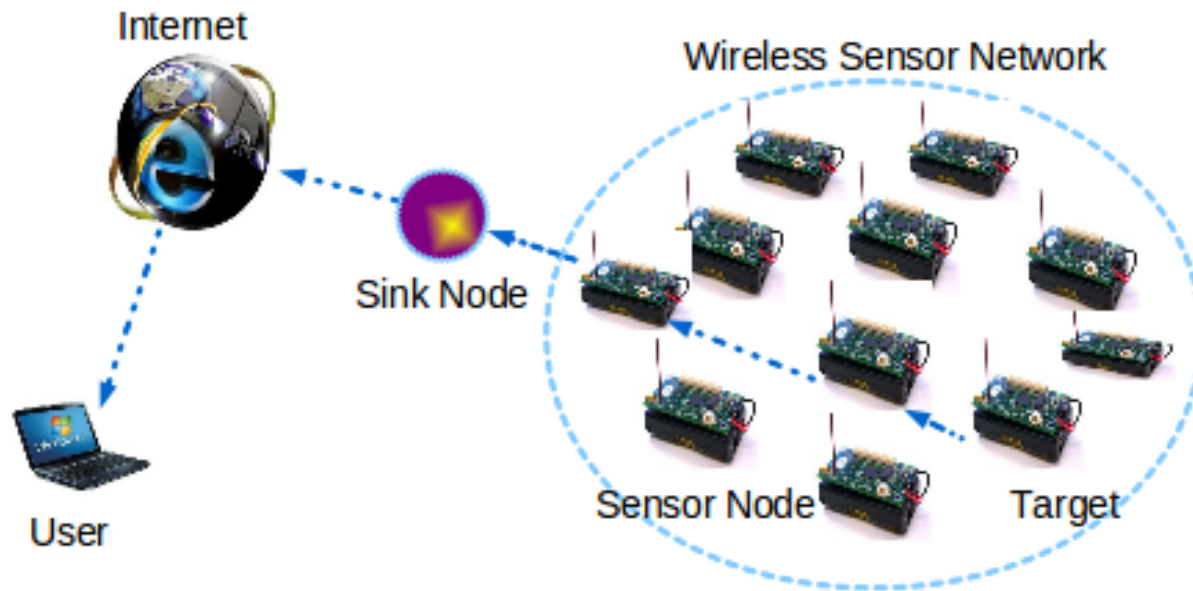
Have you seen “group testing”?

- We shall find a measurement $t \times n$ matrix A with as less rows (minimizing t) as possible such that after t projections the support set of the sparse signal can be obtained.
- Here we only consider the Boolean version $(0, 1)$ of CS .
- *Noticed that we use non-adaptive group testing algorithms for compressed sensing.*

Randomized 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 find a bunch of items which are not defective. (See it?)

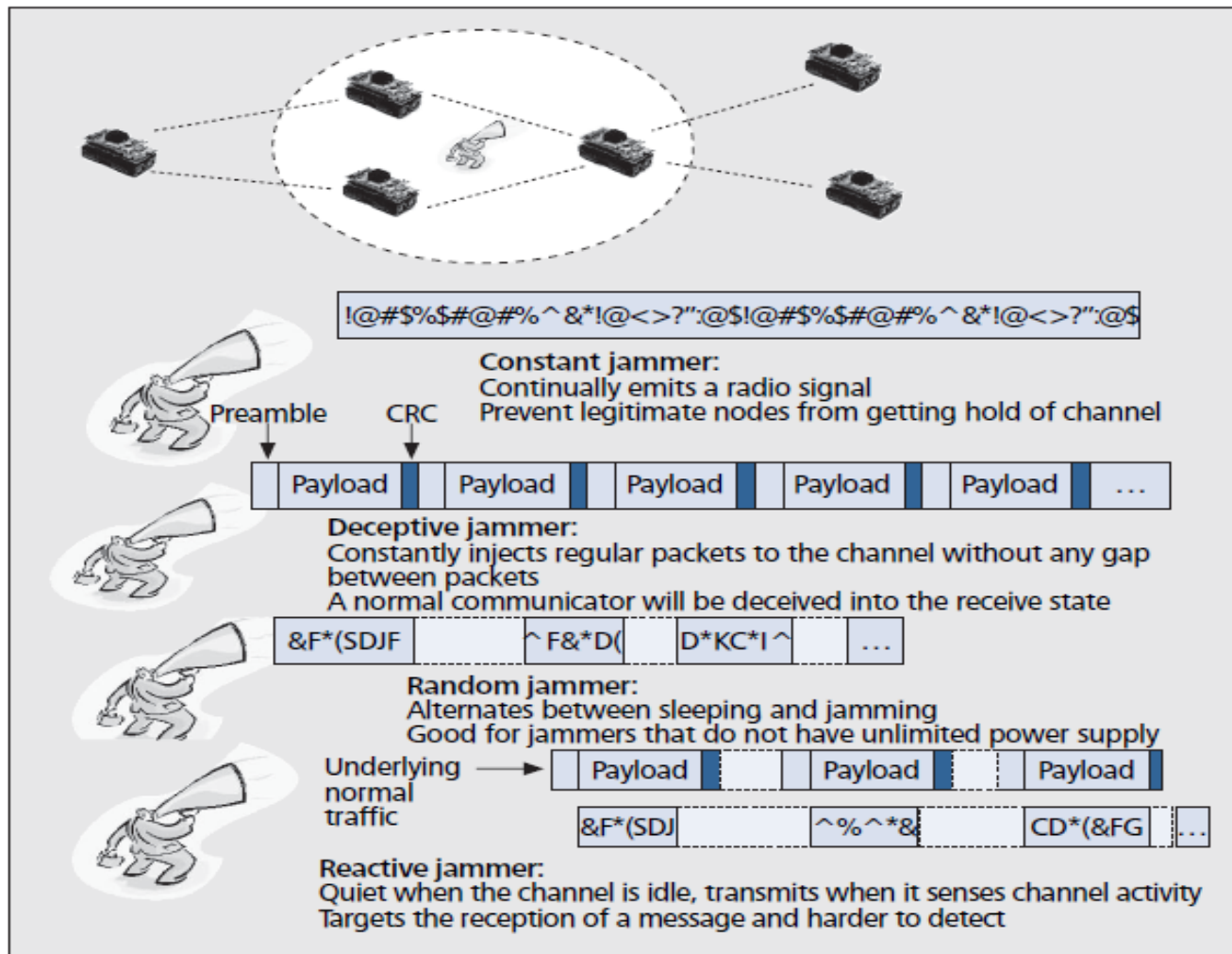
Wireless Sensor Network



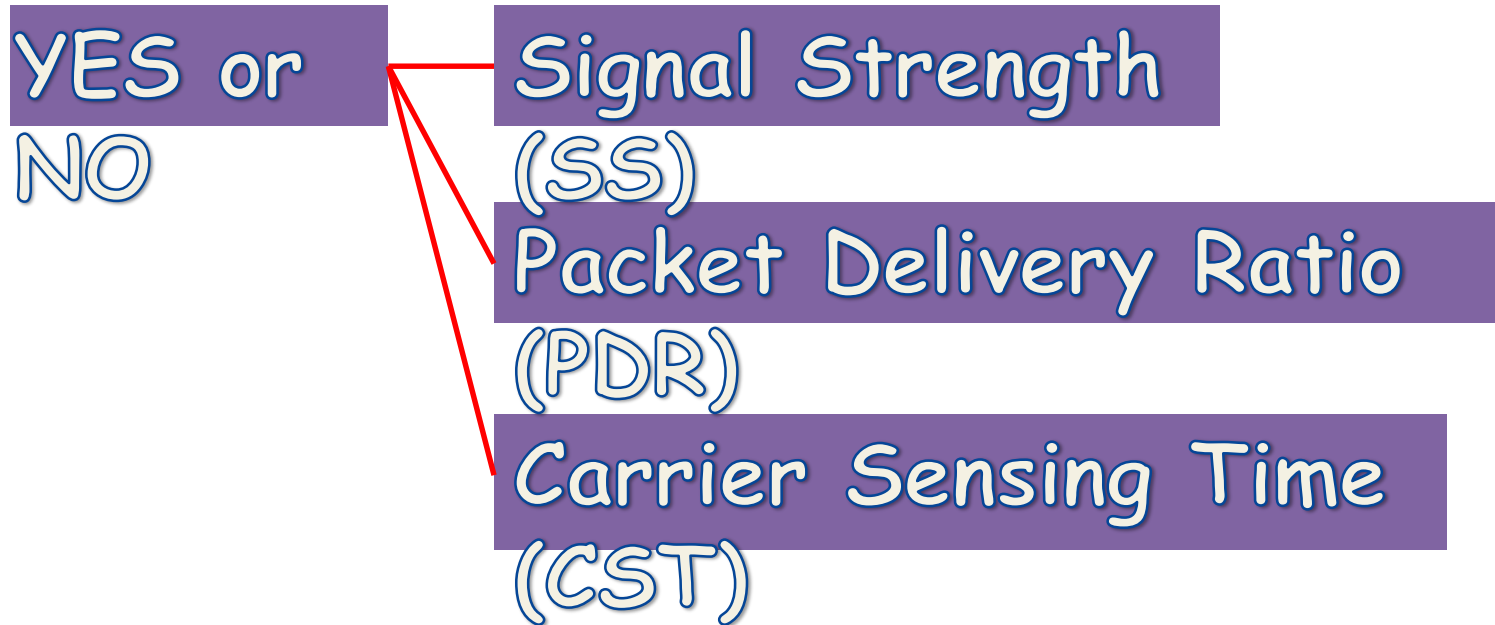
Types of Jammer

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

Jammers

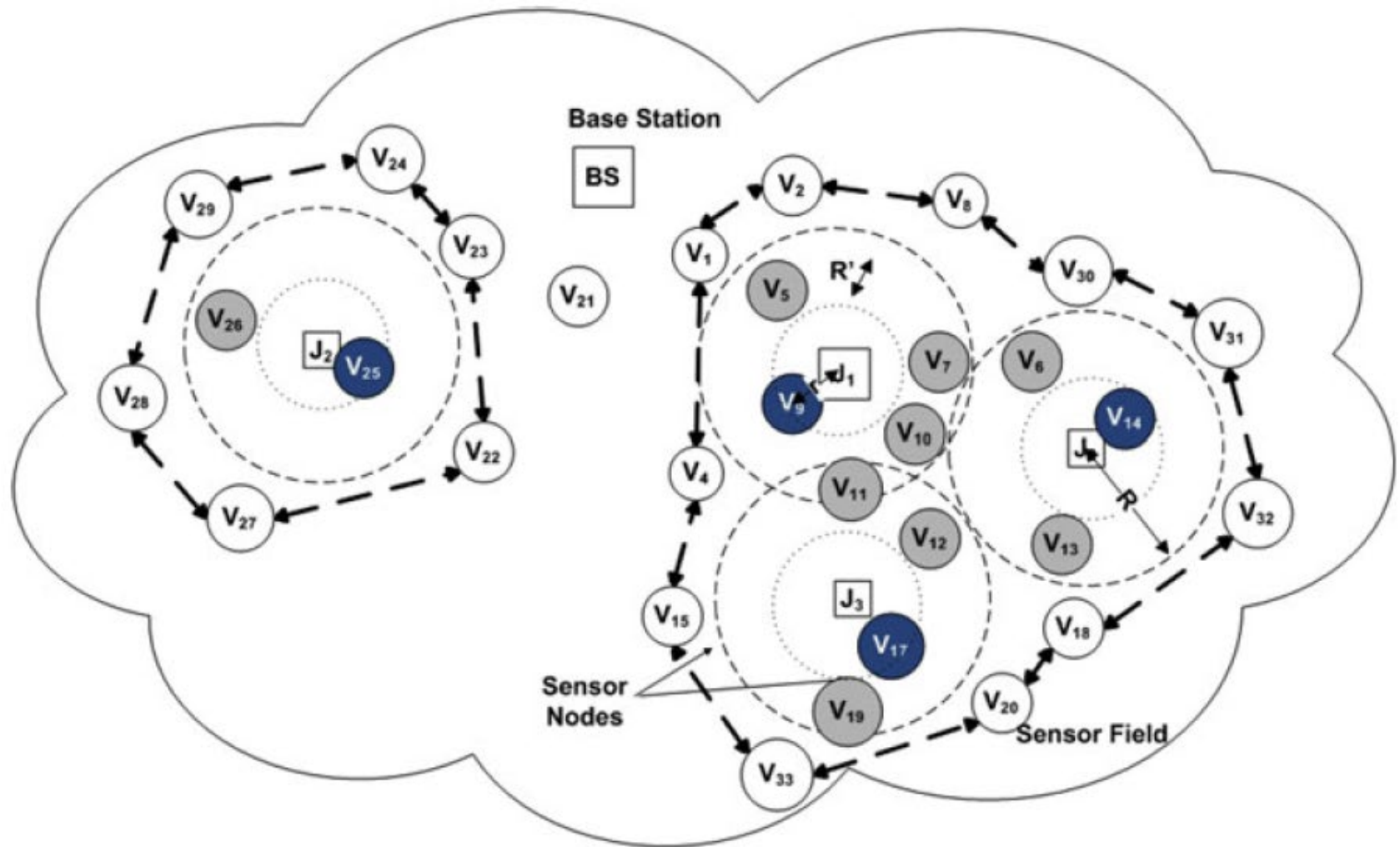


Detection for Jammers



Further Information

Basic Model



Identifying Trigger Nodes

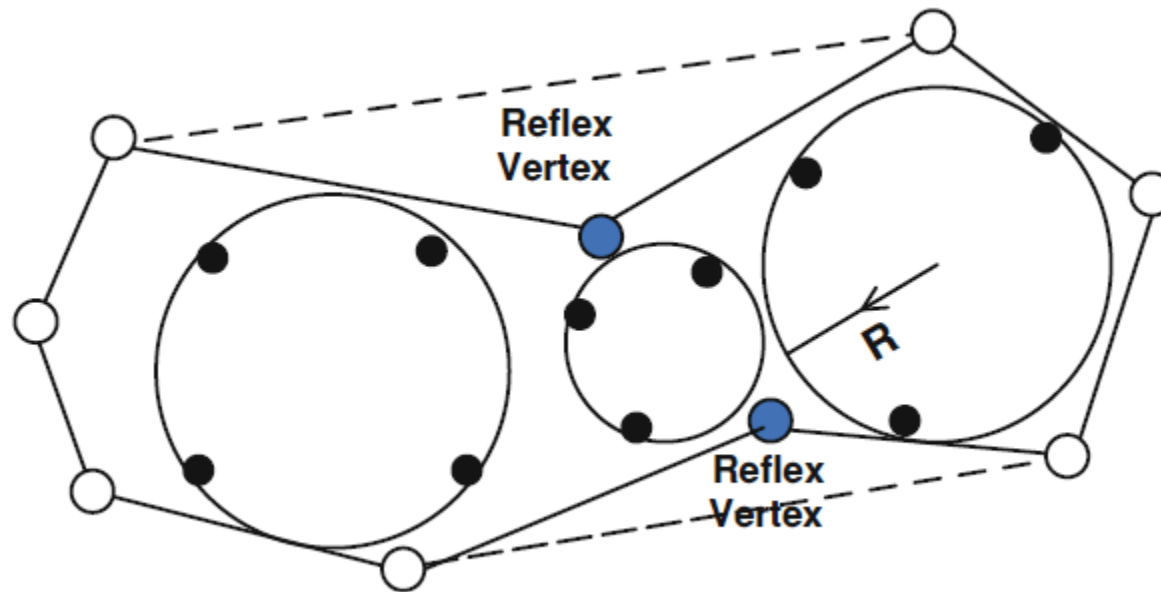
- Preprocessing

The diagram shows the word 'Preprocessing' with a red dot at its end. Two red lines branch out from this dot to the left sides of two blue rectangular boxes. The top box contains the text 'Node Classification' and the bottom box contains the text 'Jamming Range Estimation'.

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

Jamming Range Estimation

- Dense-jammer case



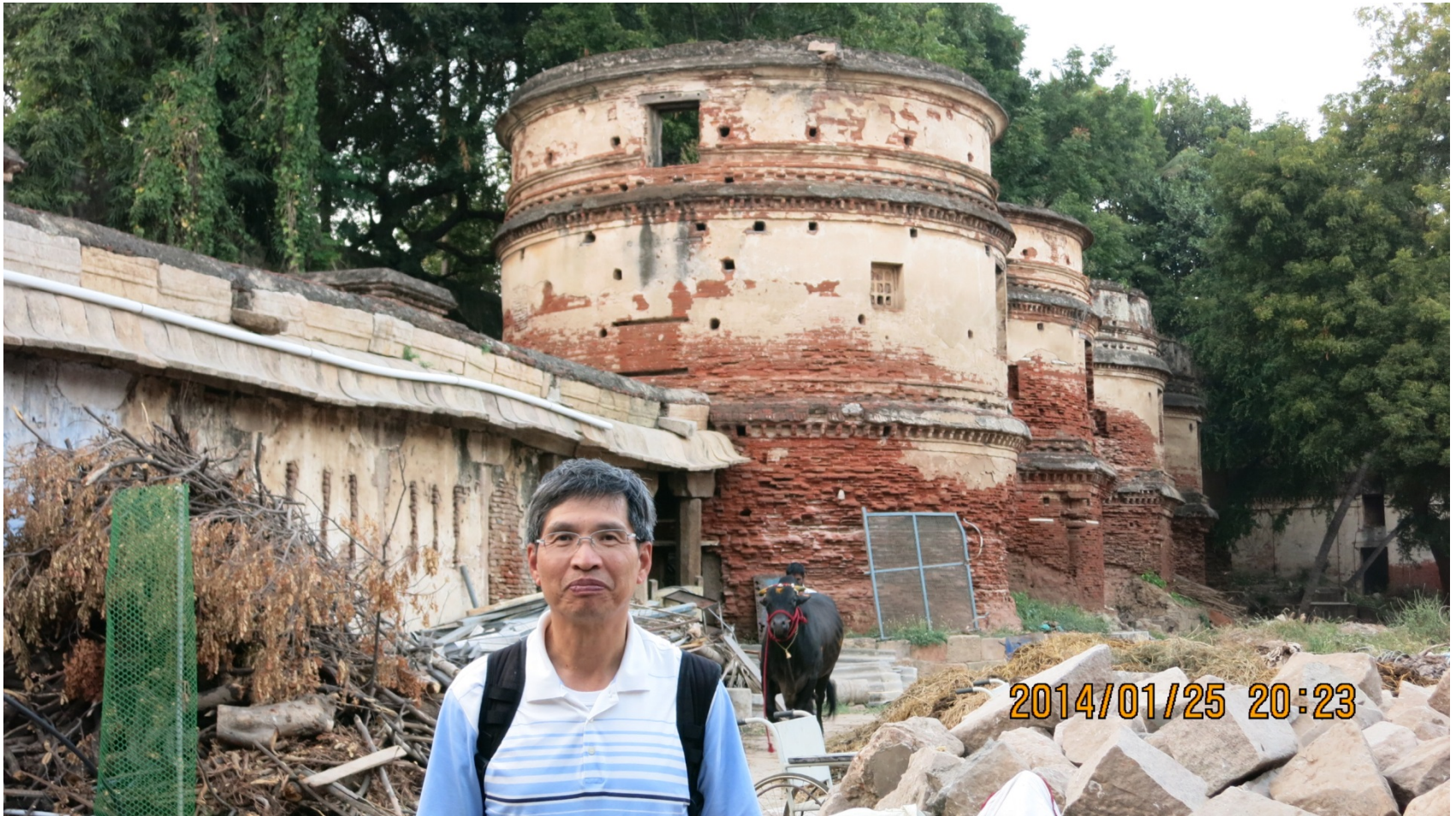
Network Security

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

Fingerprinting for Multimedia

- *Digital fingerprinting* is a technique for identifying users who use multimedia content for unintended purposes, such as redistribution.
- These fingerprints are typically *embedded* into the content using *watermarking* techniques that are designed to be robust to a variety of attacks.

Where are they?



Anti-collusion

- A cost-effective attack against such digital fingerprints is *collusion*, where several differently marked copies of the same content are combined to disrupt the underlying fingerprints.
- 集体犯案
- How can we design a “code” to catch them if it does happen? *Again, group testing works!*

Keep Moving Forward!

