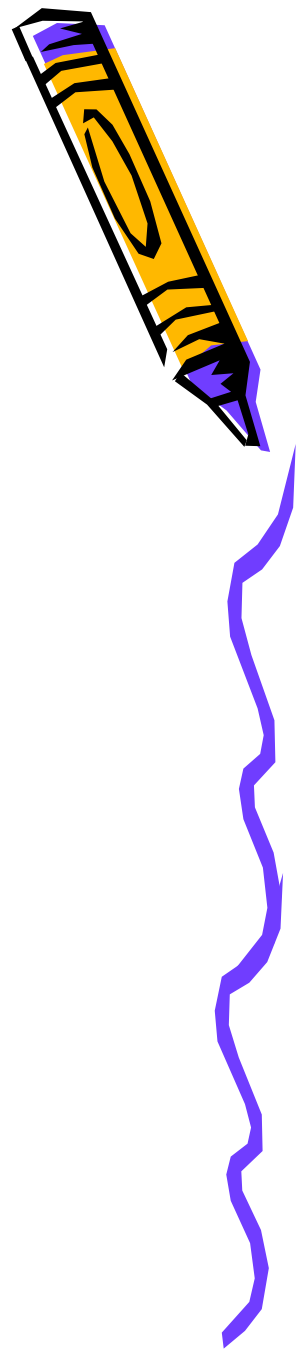
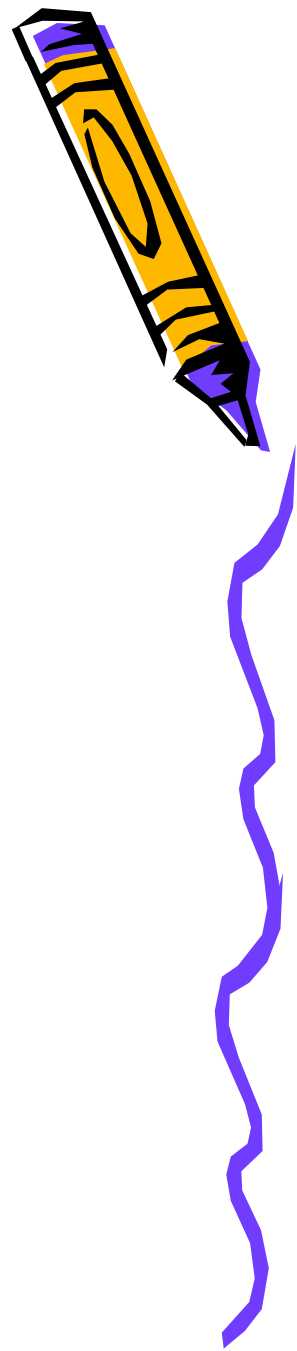


Sequencing DNA

- 交通大學應用數學系
- 傅恆霖

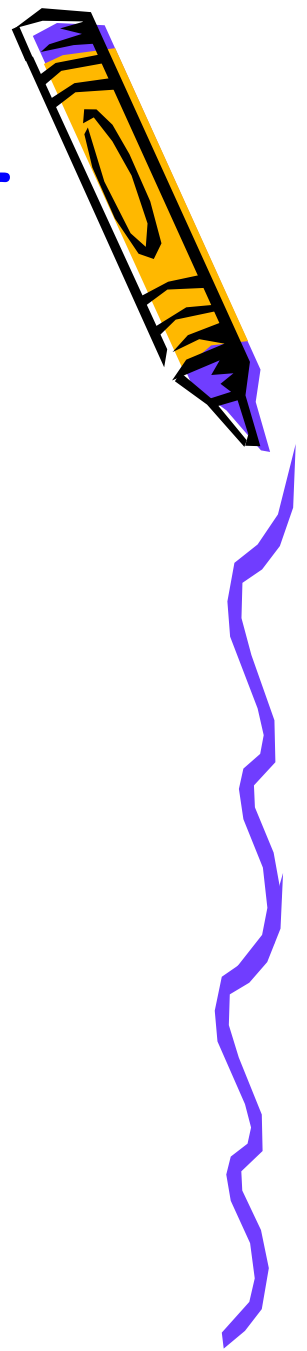


Human Genome Project





Human Genome Project



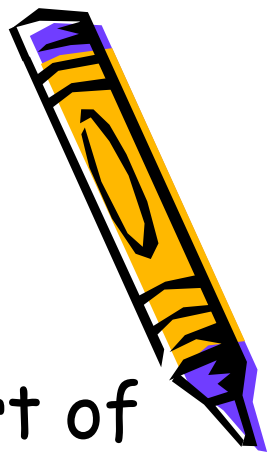
Goals:

- identify all the approximate 30,000 genes in human DNA,
- determine the sequences of the 3 billion chemical base pairs that make up human DNA,
- store this information in databases,
- improve tools for data analysis,
- transfer related technologies to the private sector, and
- address the ethical, legal, and social issues (ELSI) that may arise from the project.

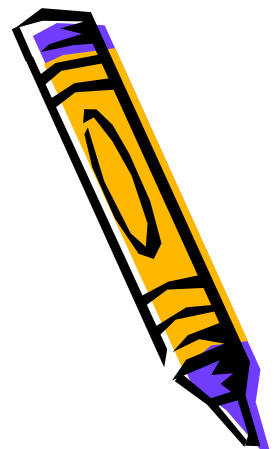


Milestones

1. 1990: Project initiated as joint effort of U.S. Department of Energy and the National Institutes of Health
2. June 2000: Completion of a working draft of the entire human genome
3. February 2001: Analyses of the working draft are published
4. April 2003: HGP sequencing is completed and Project is declared finished two years ahead of schedule

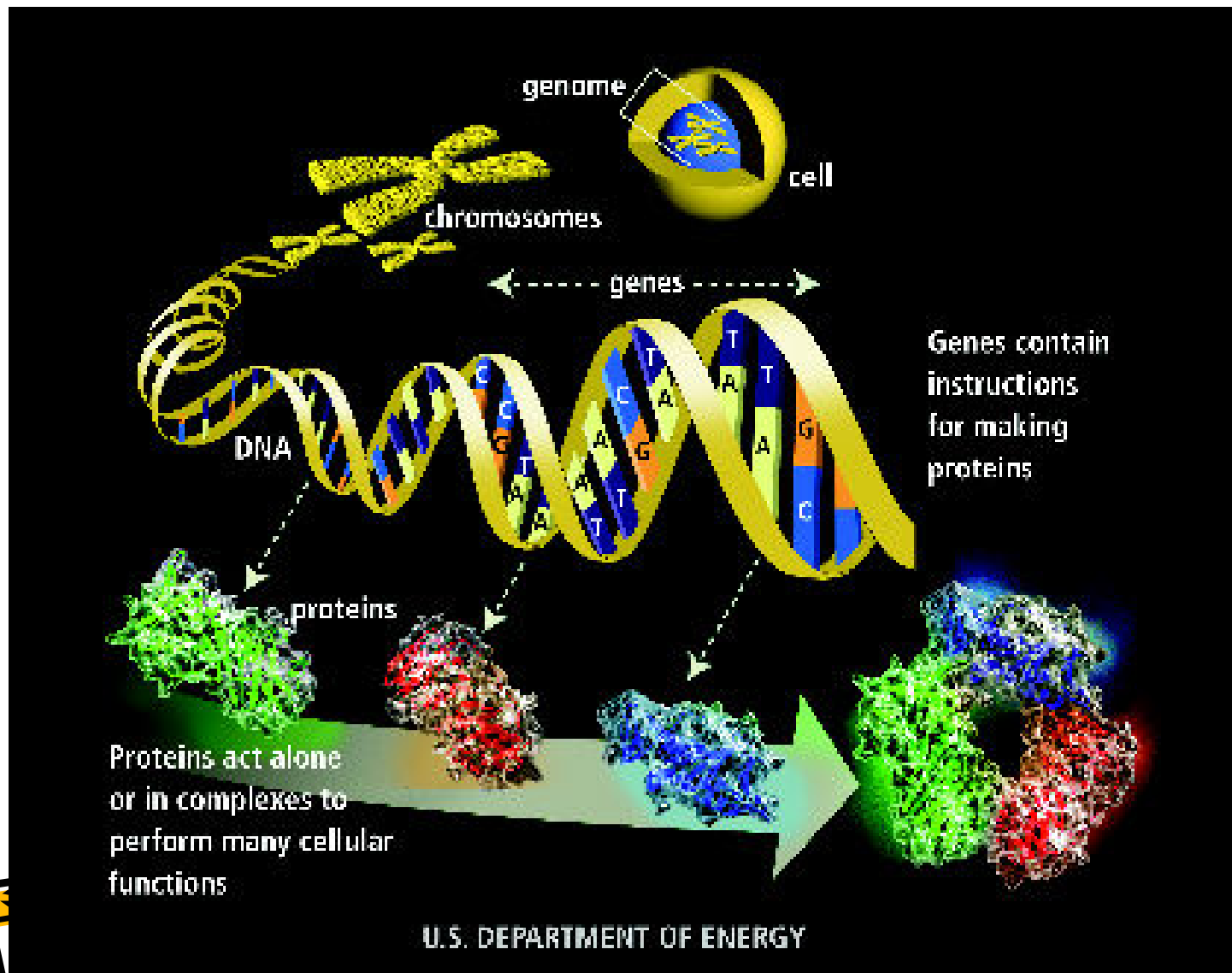


How does the human genome stack up?



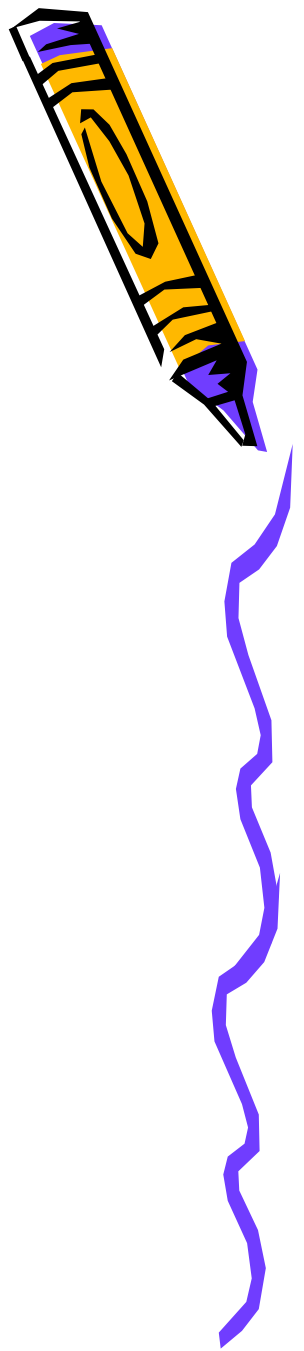
Organism	Genome Size (Bases)	Estimated Genes
Human (人類)	3 billion	30,000
Laboratory mouse (白老鼠)	2.6 billion	30,000
Mustard weed (<i>A. thaliana</i>)	100 million	25,000
Roundworm (<i>C. elegans</i>)	97 million	19,000
Fruit fly (果蠅)	137 million	13,000
Yeast (酵母菌)	12.1 million	6,000
Bacterium (大腸桿菌)	4.6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





去氧核糖核酸

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



Proteins (蛋白質)

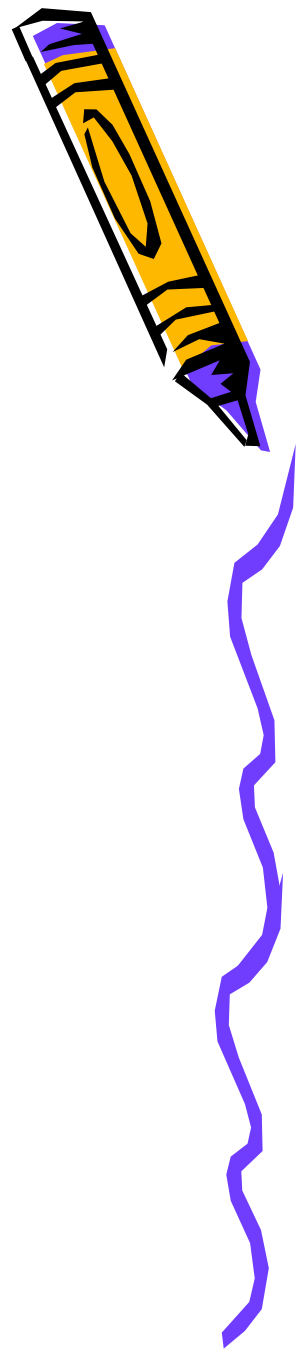


- The twenty *amino acids* commonly found in proteins:
 - **A** **Ala** Alanine
 - **C** **Cys** Cysteine
 - **D** **Asp** Aspartic Acid
 - **E** **Glu** Glutamic Acid



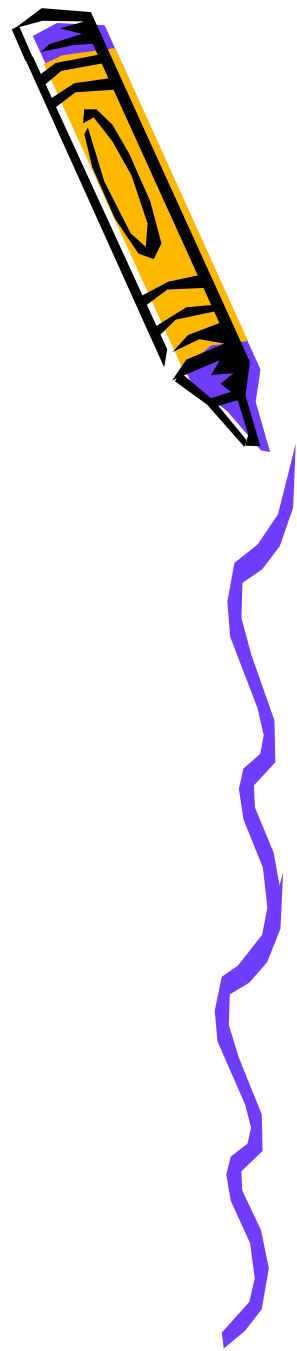
Continued ...

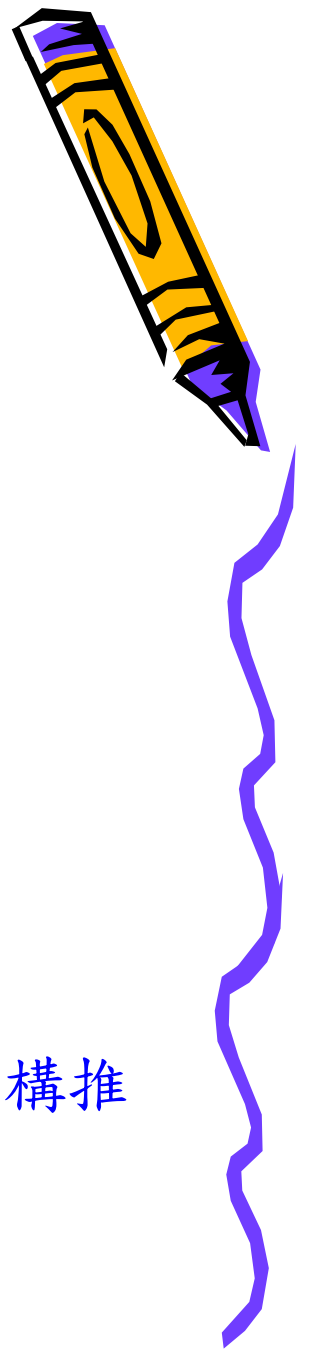
- F Phe Phenylalanine
- G Gly Glycine
- H His Histidine
- I Ile Isoleucine
- K Lys Lysine
- L Leu Leucine
- M Met Methionine
- N Asn Asparagine
- P Pro Proline



Continued ...

- Q Gln Glutamine
- R Arg Arginine
- S Ser Sreine
- T Thr Threonine
- V Val Valine
- W Trp Trptophan
- Y Tyr Tyrosine





- 計算分子生物學簡要

- 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 (Sequencing)



• 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)。



Strings

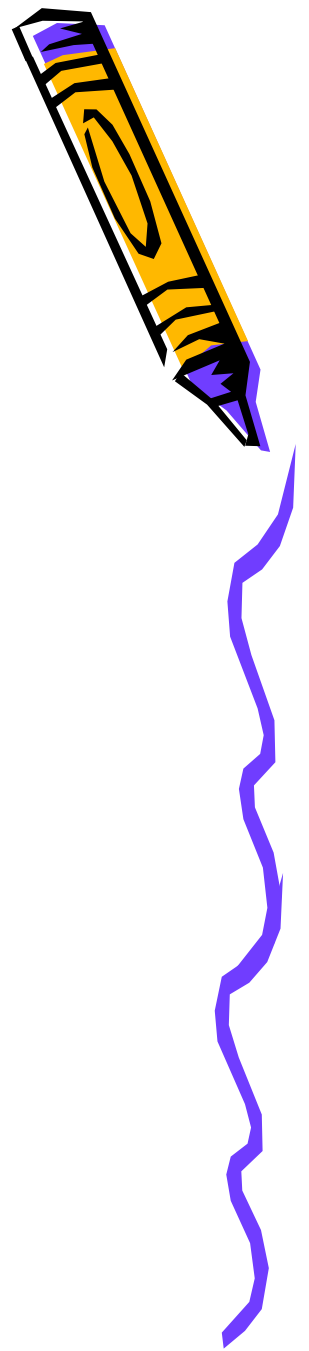


- A *string* is an ordered succession of characters or symbols drawn from a finite set called the *alphabet*.
- Most of the time, we use either the DNA alphabet {A, C, G, T} or the 20 amino acids alphabet {A, C, D, E, ... }.
- String $\equiv$ Sequence



Limitation

- 定序儀壹次只能定700個鹼基對左右，必須組合才能得到更長的序列
- ... ATCGGATGCCTTAG
- CCTTAGCAACTGA...
- ... ATCGGATGCCTTAGCAACTGA ...
- 利用序列兩端
- 的相似性可將
- 兩序列連起來
- 組合序列就像是
玩拼圖遊戲



Similarity



- How similar are two sequences?

Definition 1 (Alignment)

An alignment of two sequences S and T is a pair of sequences (S', T') obtained by insertion of spaces in S and T (respectively) such that (1) S' and T' are of the same length and (2) for each corresponding position, at most one of them is a "space".



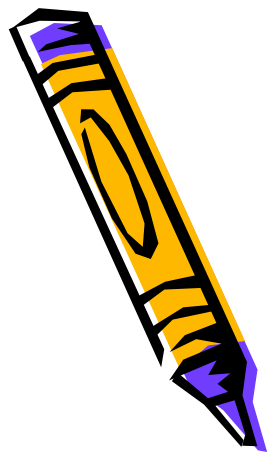
Example and explanation

S: GACGGATTAG → S': GA-CGGATTAG

T: GATCGGAATAG → T': GATCGGAATAG

◎ 對齊方式條件 (Alignment)

- (1) 二序列有不同長度，需要用「空格」填補，使二序列長度一樣長。
- (2) 二個序列不可「空格 對 空格」。
- (3) 空格允許插入在字串的最前端或最尾端。



Score of Alignment



- Definition 2

The score of the pair (S, T) of aligned sequences, denoted by $\sigma(S, T)$ is the sum of $\sigma(S(i), T(i))$ where $\sigma(x, y)$ is equal to $p > 0$ if $x = y$, $q < 0$ if x and y are different, and $r < 0$ if one of x and y is a "space".

Here $S(i)$ (respectively $T(i)$) is the i th character (may be a space) of S (resp. T).



Scores

- There are many different ways to define the score of two aligned sequences.
- The definition provided above is just an example!
- Suitable scoring system gives better result of sequencing.



Optimal alignment



- Definition 3 (Similarity of S and T)

The similarity of S and T denoted by $\text{Sim}(S, T)$ is equal to the maximum value of $\sigma(S', T')$ among all possible alignments (S', T') of (S, T) .

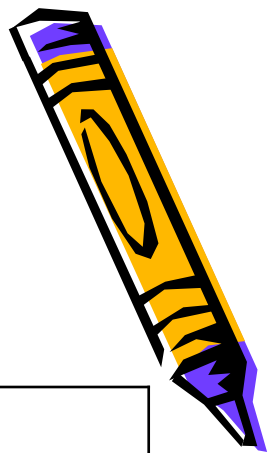
For example, $\text{Sim}(AGC, AAAC) = -1$ if we let $p = 1$, $q = -1$ and $r = -2$.

The pair (S', T') which attend the similarity value is an optimal alignment.



Dynamic Algorithm

	-	A	G	C
-	0	-2	-4	-6
A	-2	1	-1	-3
A	-4	-1	0	-2
A	-6	-3	-2	-1
C	-8	-5	-4	-1



How to transform one sequence to another?



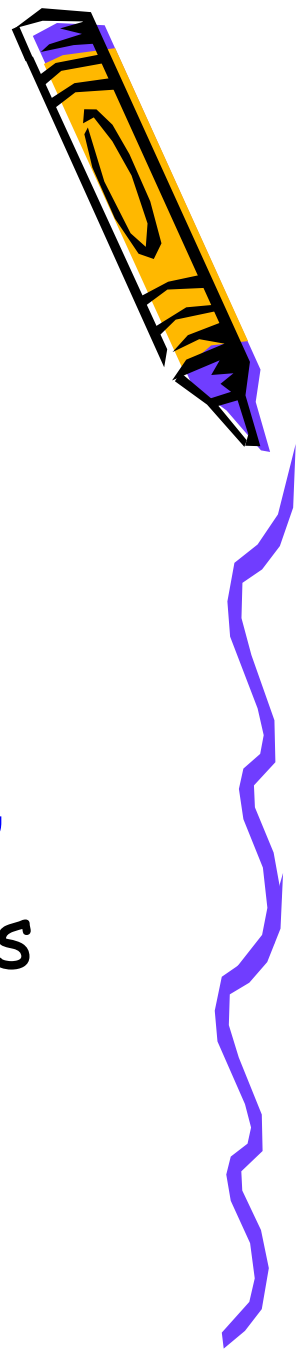
- Definition 4 (Edit distance)

The edit distance between two sequences is the minimum number of edit operations needed to transform one sequence into another, where the operations are (1) insertion of a character, (2) deletion of a character, and (3) substitution of a character for another one.

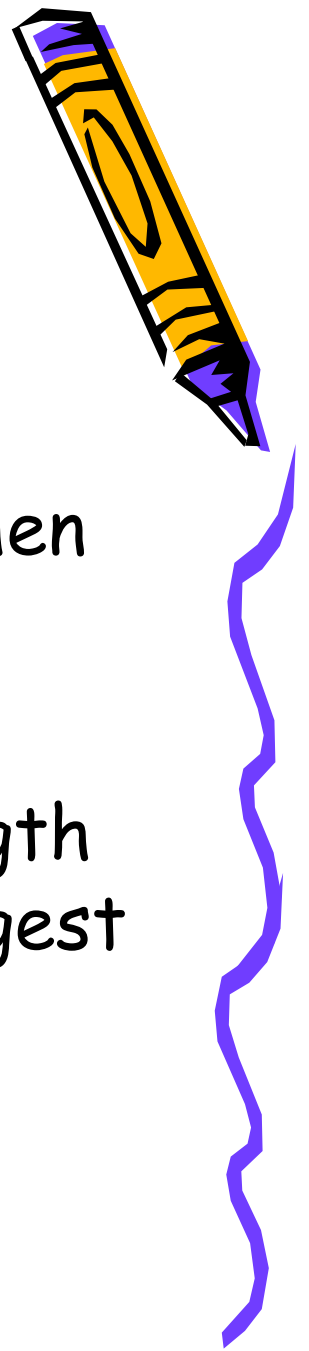


Other operations

- In real world applications, we may also reverse a substring.
- For example, AAGTGCTA and AACGTGTA.
- If we count this operation as “one” operation, then the edit distance is going to be smaller.



Longest common subsequence problem



- If the edit operations are only from insertion and deletion of characters, then we have the edit distance of V and W ,

$$d(V, W) = n + m - 2s(V, W)$$

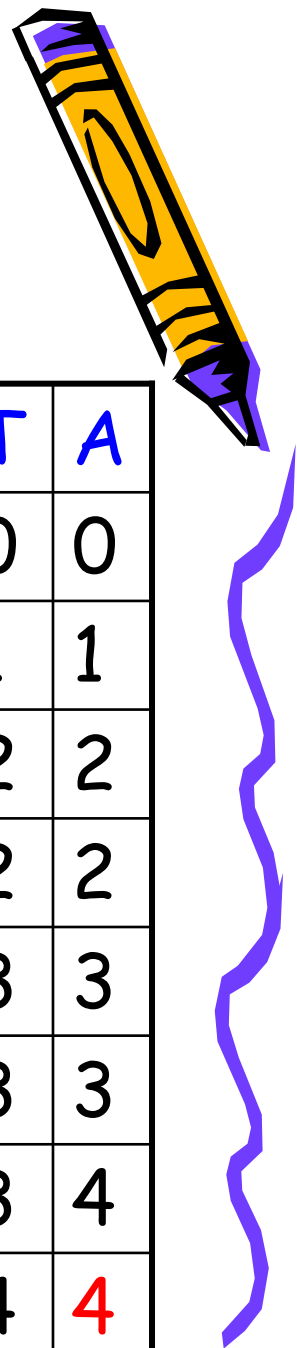
where n is the length of V , m is the length of W and $s(V, W)$ is the length of a longest common subsequence of V and W .



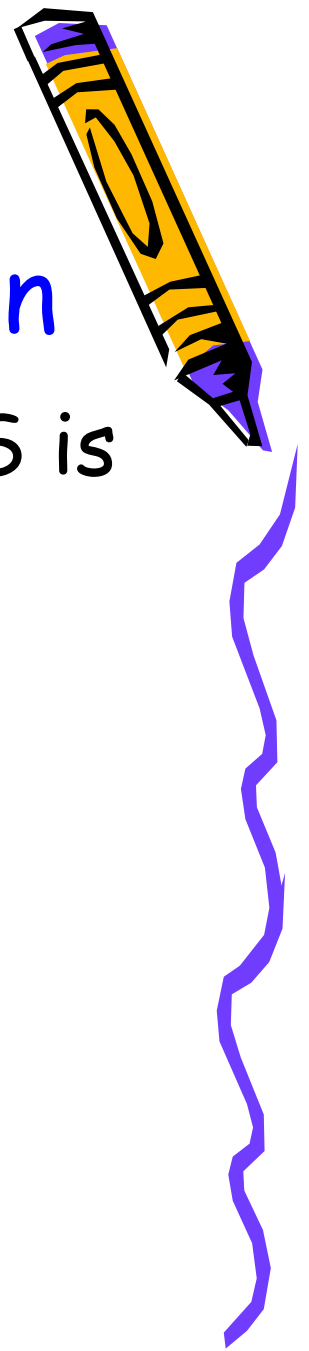
Dynamic algorithm for LCS problem

- Let $V = \text{ATCTGAT}$ and $W = \text{TGCATA}$.
- $S(V, W) = 4$

	-	T	G	C	A	T	A
-	0	0	0	0	0	0	0
A	0	0	0	0	1	1	1
T	0	1	1	1	1	2	2
C	0	1	1	2	2	2	2
T	0	1	1	2	2	3	3
G	0	1	2	2	2	3	3
A	0	1	2	2	3	3	4
T	0	1	2	2	3	4	4



Longest Increasing subsequence of a permutation

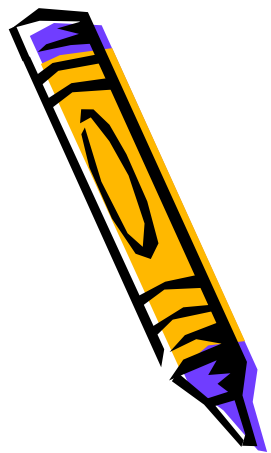


- A permutation of a nonempty set S is a bijection from S onto S .
- It is well known that the set of permutations defined on the set $\{1, 2, 3, \dots, n\}$ and the operation “composition” create a **symmetric group** S_n which has $n!$ elements.



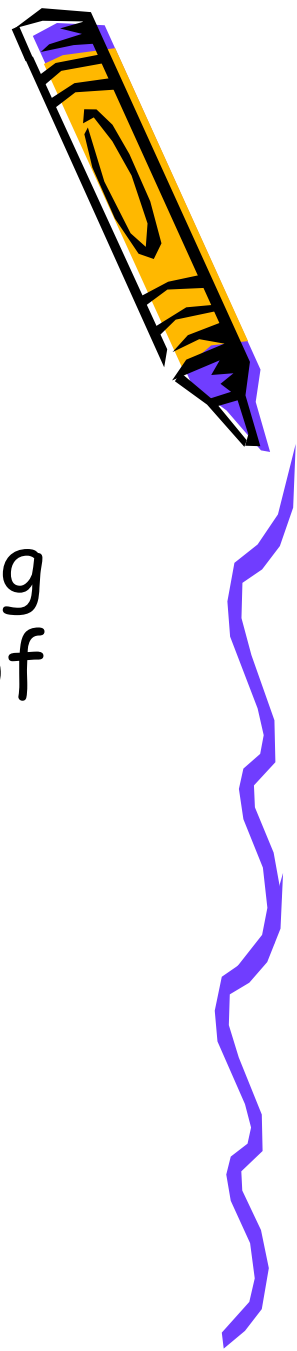
Permutation

- An element π of S_n can be denoted by $(x_1 x_2 x_3 \dots x_n)$ where $\pi(i) = x_i$ for $i = 1, 2, 3, \dots, n$.
- For example, $(7 \ 2 \ 8 \ 1 \ 3 \ 4 \ 10 \ 6 \ 9 \ 5)$ is a permutation which maps 1 to 7, 2 to 2, 3 to 8, ... etc.
- The length of **a** longest increasing subsequence of the above permutation is **5**.

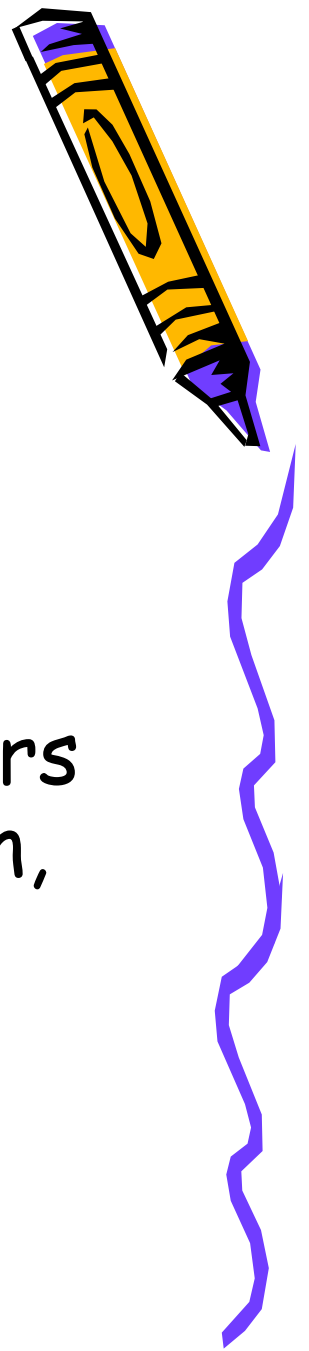


Observation

- Finding the longest increasing subsequence of a permutation $(x_1 x_2 x_3 \dots x_n)$ is equivalent to finding the **longest common subsequence** of this permutation with the identity permutation $(1\ 2\ 3 \dots n)$.
- So, a dynamic algorithm works (as mentioned above)!



Better idea?



- More mathematics!

Definition 5 (Partition of integer)

A partition of an integer n is a sequence of non-decreasing integers $a_1, a_2, \dots, a_k$ such that their sum is n , denoted by $(a_1, a_2, \dots, a_k) \vdash n$.

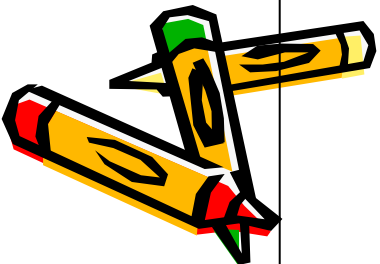
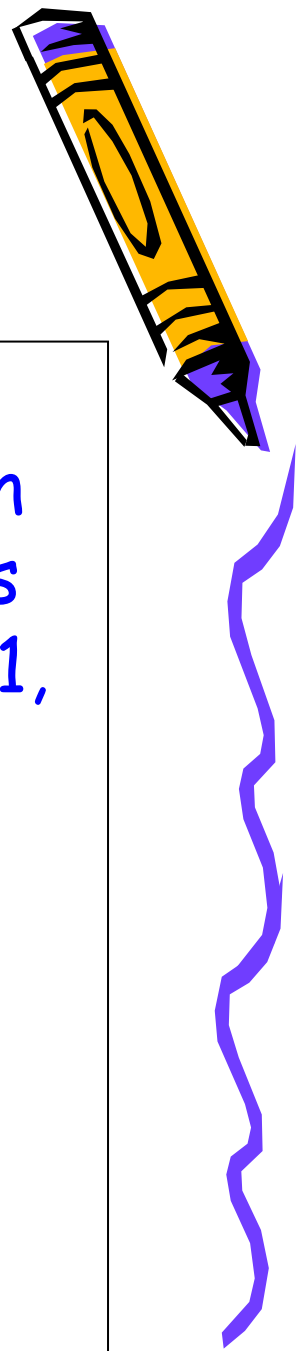
For example $(4, 2, 2, 1) \vdash 9$.



Young Diagram

- The Young diagram of shape $\mathbf{a} = (a_1, a_2, \dots, a_k) \vdash n$ is an array of n cells which has k left-justified rows with row i containing a_i cells for $i = 1, 2, \dots, k$.
- For example:
 $(4, 2, 1, 1) \vdash 8$

1	2	5	8
4	7		
6			
9			



Young Tableaux



- Definition 6

A Young tableau of shape $\mathbf{a} = (a_1, a_2, \dots, a_k)$ $\vdash n$ is an array obtained by replacing the cells of the Young diagram of $\mathbf{a}$ with the numbers $1, 2, \dots, n$ bijectively. A tableau is **standard** if its rows and columns are increasing sequences (from the left and top respectively).

- The example of last slide is not a standard Young tableau.



Permutation and Tableaux



- We can use a pair of standard Young tableaux to represent a permutation, called (P, Q) pair for the permutation π .
- The algorithm to obtain such a correspondence is called an **RSK algorithm** in honor of Robinson (1938), Schensted (1961) and Knuth (1970).



An example is worth of a thousand words!



- Let $\pi = (7\ 2\ 8\ 1\ 3\ 4\ 10\ 6\ 9\ 5)$.

• Why?

P

Q

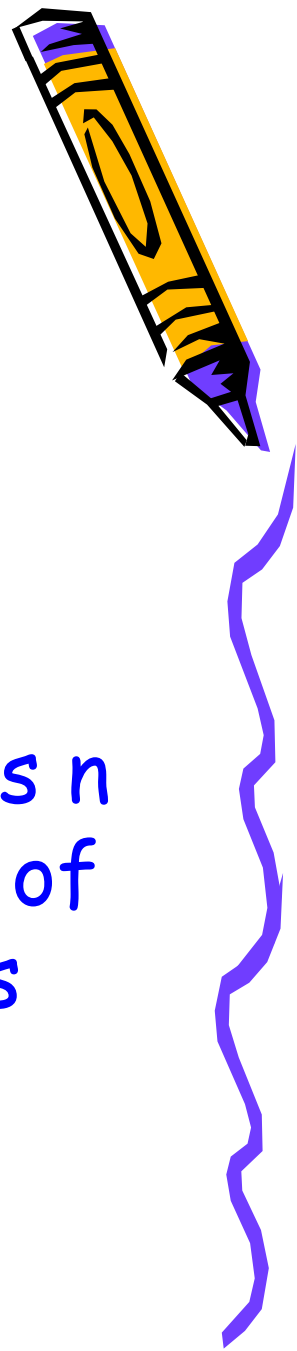
1	3	4	5	9
2	6	10		
7	8			

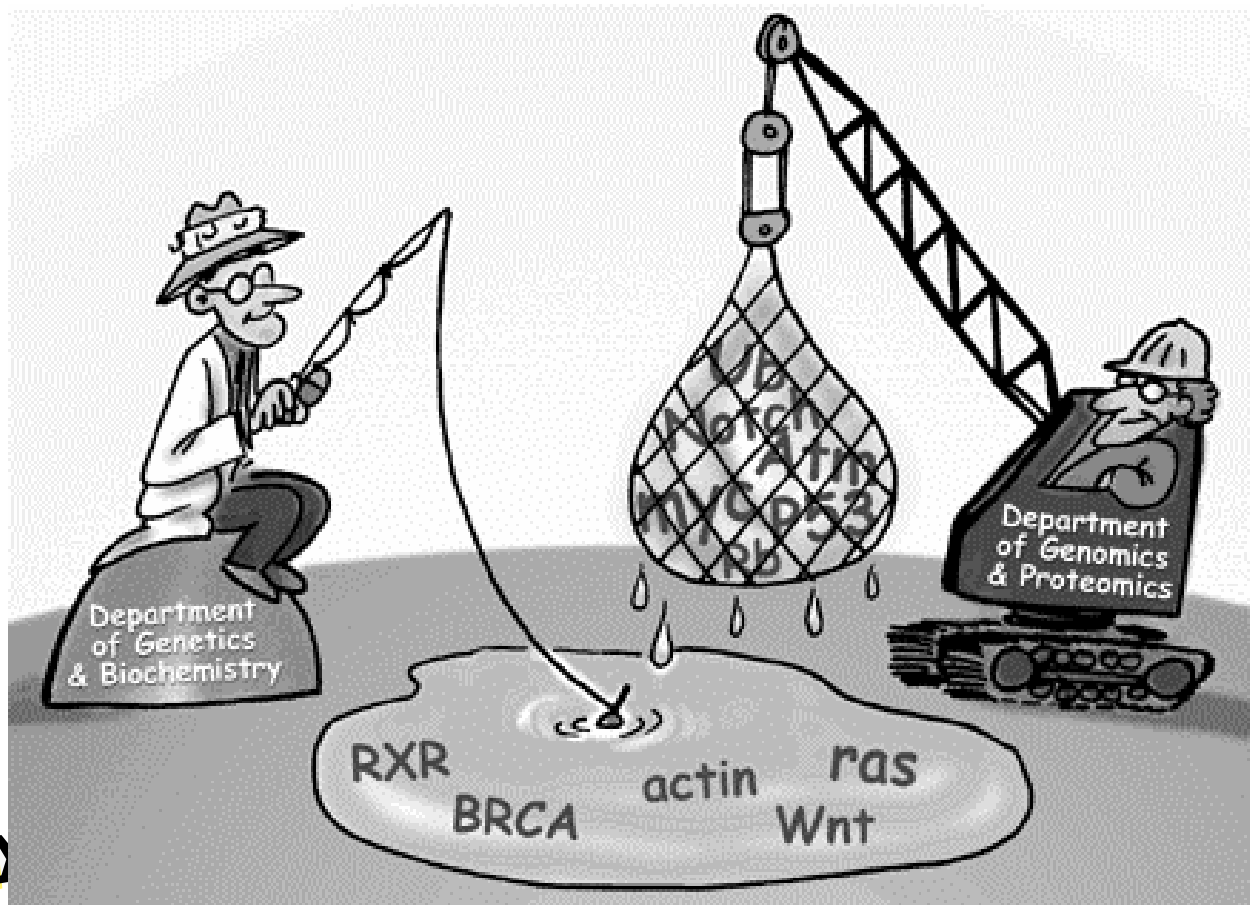
1	3	6	7	9
2	5	8		
4	10			



The length of longest increasing subsequence

- The number of columns is equal to the length of a longest increasing subsequence, see it?
- If the permutation we consider has n elements, then the average length of a longest increasing subsequence is about $2n^{1/2}$. (Believe it?)

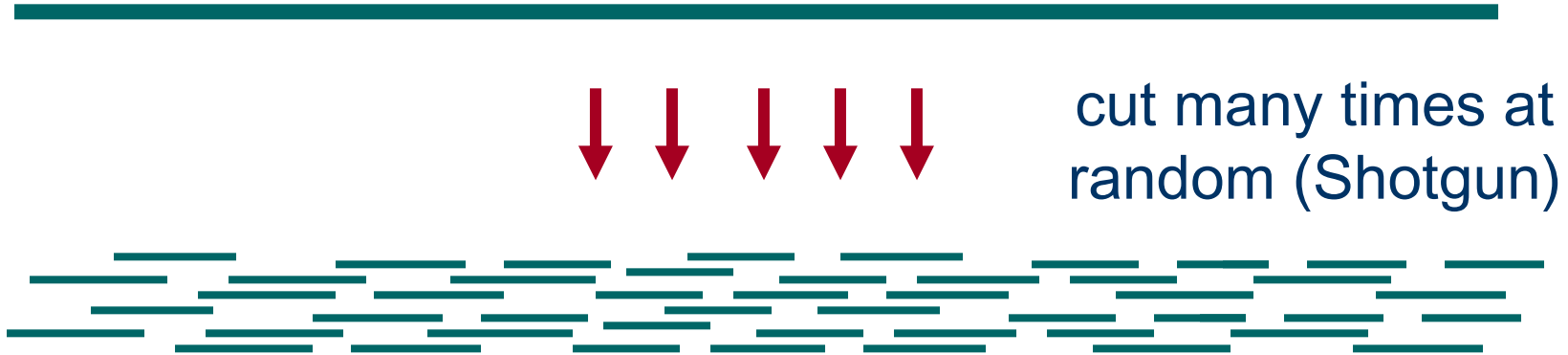




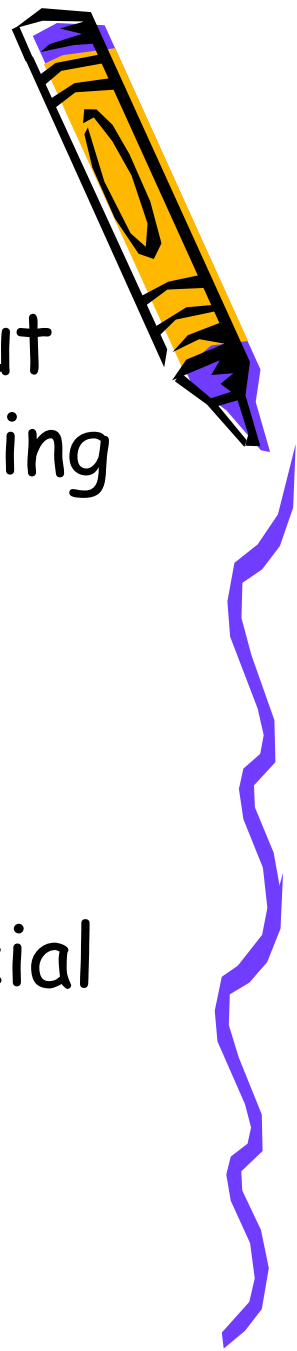
Random shotgun approach



genomic segment



Shotgun Sequencing

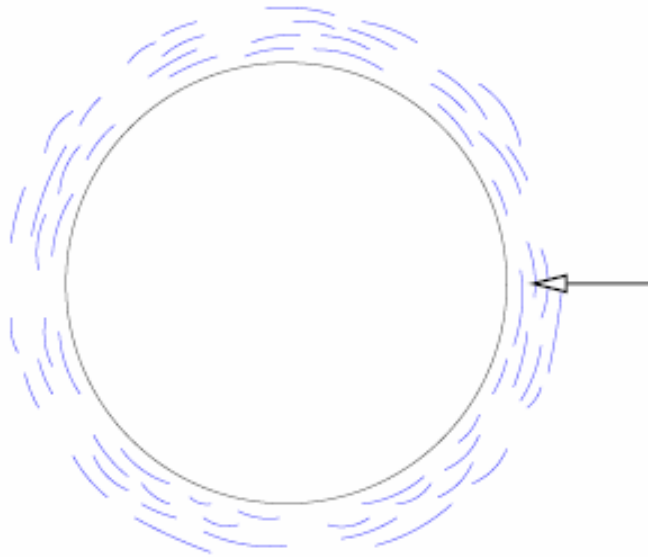


- **Shotgun sequencing** is a throughput technique resulting in the sequencing of a large number of bacterial genomes, mouse genomes and the celebrated human genomes. In all such projects, we are left with a collection of contigs that for special reasons cannot be assembled with general assembly algorithms.

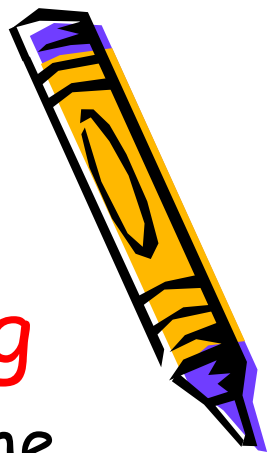
Continued ...



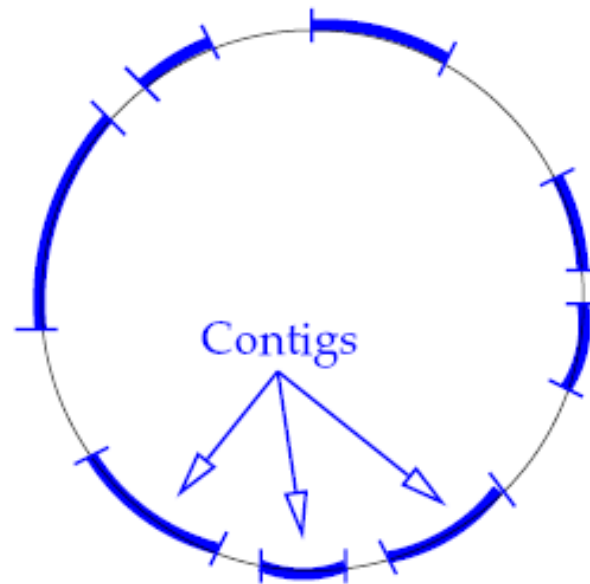
- **Whole-genome shotgun sequencing**
 - Short reads are obtained and covering the genome with redundancy and possible gaps.



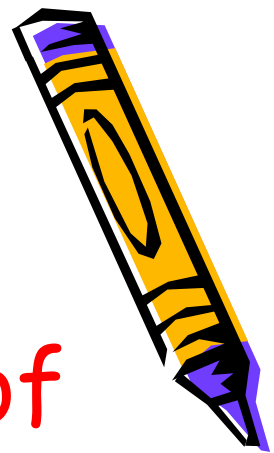
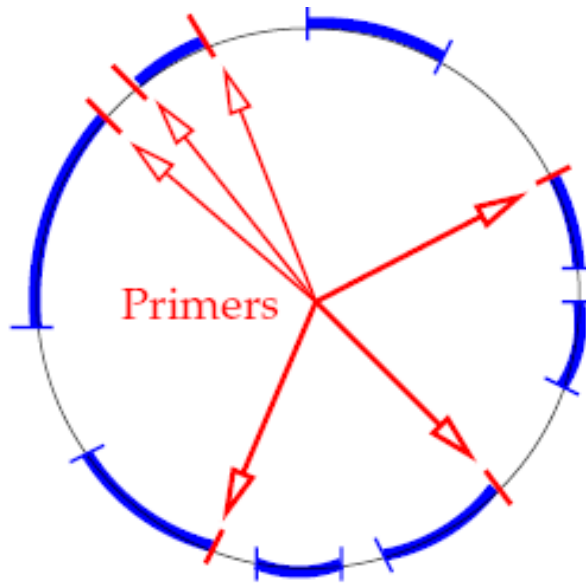
Circular genome

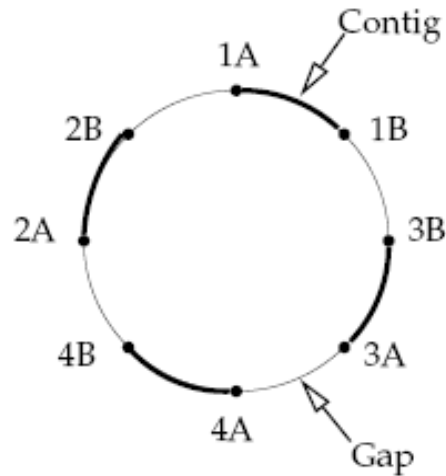


– Reads are assembled into contigs with unknown relative placement.



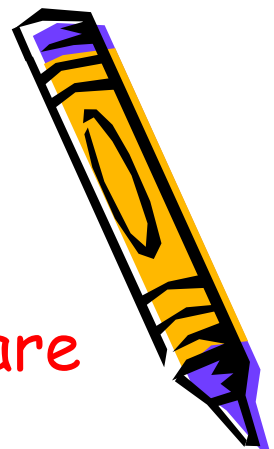
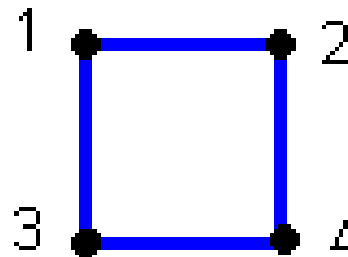
–Primers : (short) fragments of DNA characterizing ends of contigs.

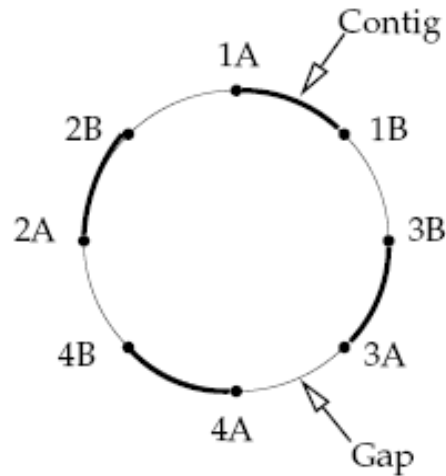




- Two primers of each contig are “mixed together”
 - Find a Hamiltonian cycle by PCRs!

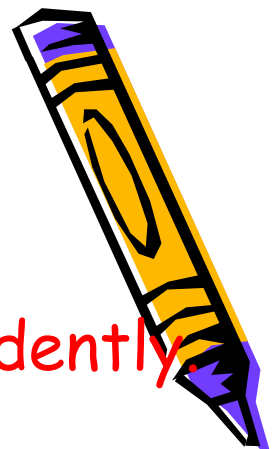
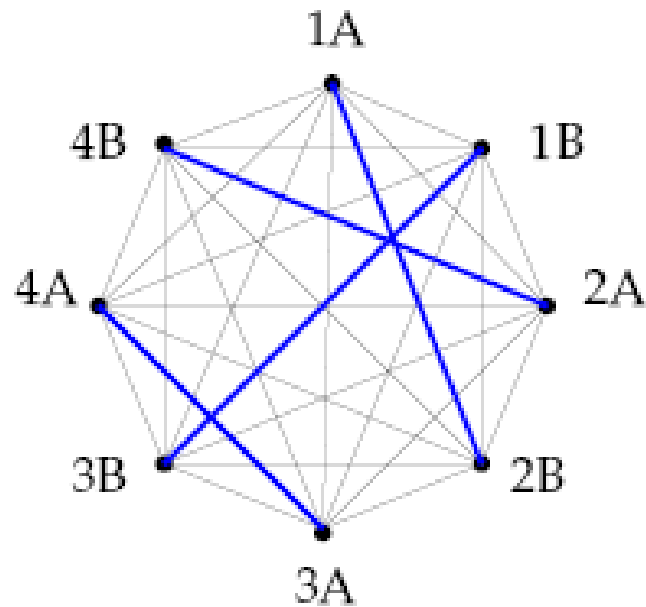
1A ——— 1B
 2A ——— 2B
 3A ——— 3B
 4A ——— 4B





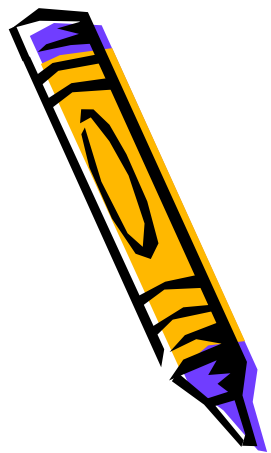
1A ——— 1B
 2A ——— 2B
 3A ——— 3B
 4A ——— 4B

- 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.



Group Testing



- Robert Dorfman's paper in 1943 introduced the field of (Combinatorial) Group Testing. The motivation arose during the Second World War when the United States Public Health Service and the Selective service embarked upon a large scale project. The objective was to weed out all **syphilitic (梅毒)** men called up for induction. However, syphilis testing back then was expensive and testing every soldier individually would have been very cost heavy and inefficient.



Formal Definitions

- 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.



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.

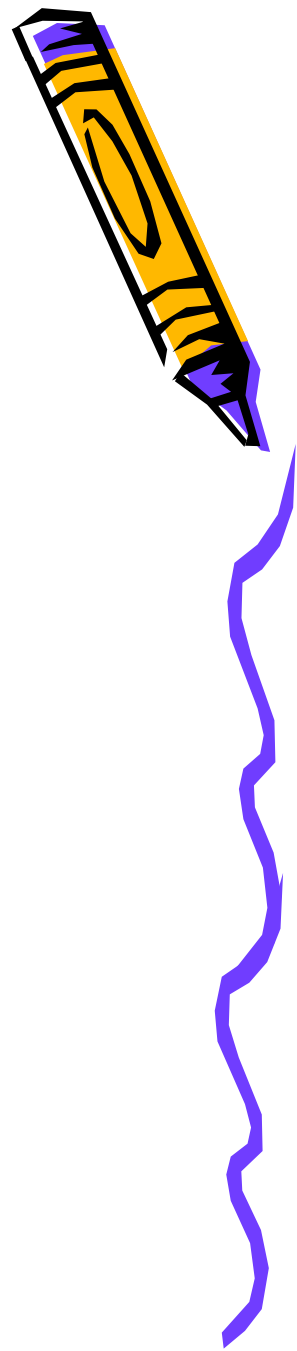


Non-adaptive Algorithm

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.



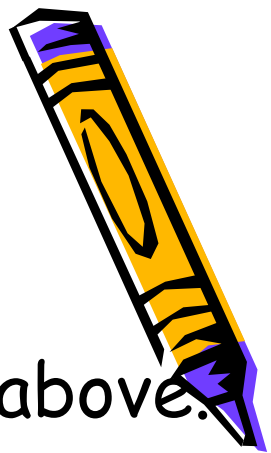
An Example

- 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

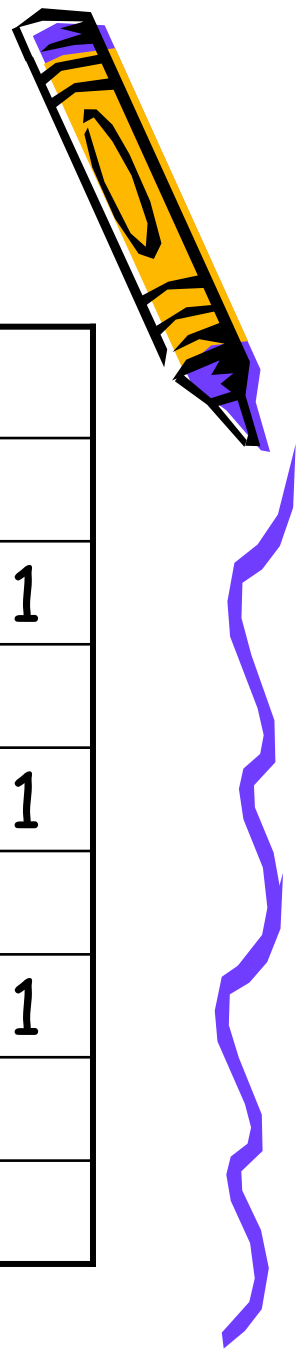
$$S_k = \{i : m_{i,k} = 1, i = 1, 2, \dots, t\}, k = 1, 2, \dots, n.$$

- 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?)



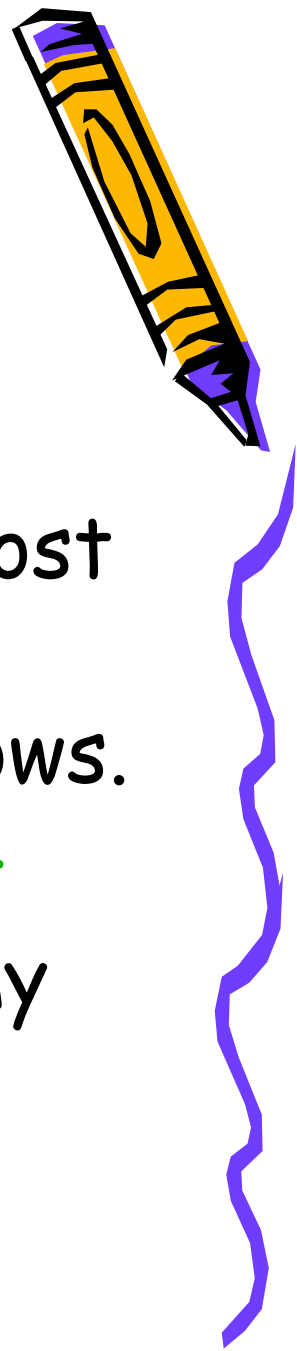
Incidence Matrix



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	



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. (?)



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.



Important Facts



- A d -disjunct matrix is also a d -separable matrix.
- *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.

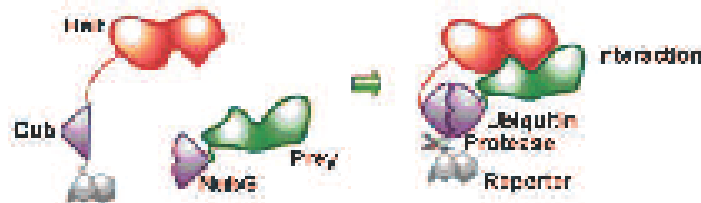
- *d -disjunct matrices have a simple decoding algorithm, namely, a column is positive if and only if it does not appear in a negative row.*



More group testing models

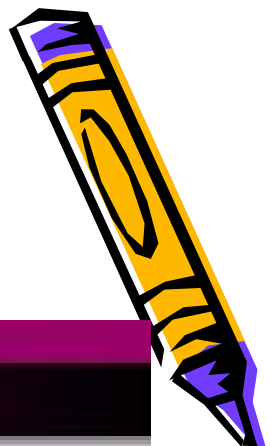
Complex Model

- ✳ Application: Protein-to-protein interaction (Lappe and Holm, 2004):



<http://www.mdc-berlin.de>

- Signals from the exterior of a cell are mediated to the insider of that cell by protein-to-protein interaction of the signaling molecules.
- Creating a protein-protein interaction network for Alzheimer's disease is vital.

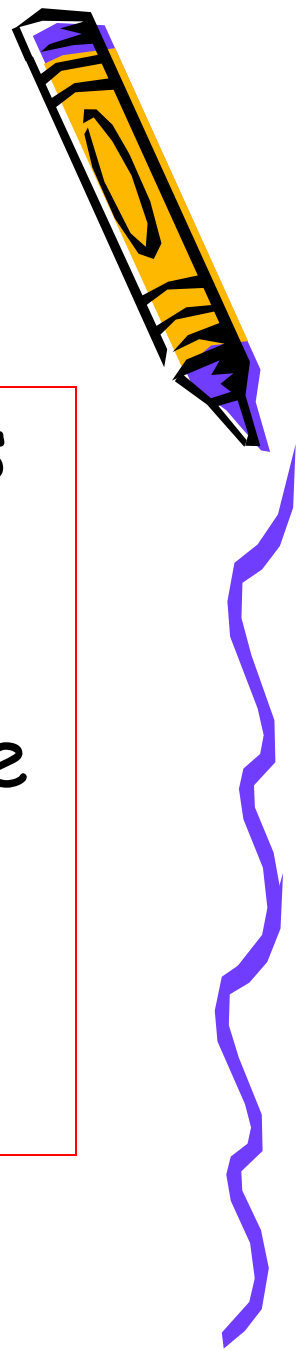


Inhibitor Model

- ✦ An *inhibitor* is a third type of clones whose existence may cancel the effect of positive clones (Farach et al., 1997).
- ✦ Application:
 - In *drug discovery*, certain compound can block the detection of potent compounds (Xie et al. 2001).
 - *Enzyme inhibitors* are molecules that interact in some way with the enzyme to prevent it from working normally.

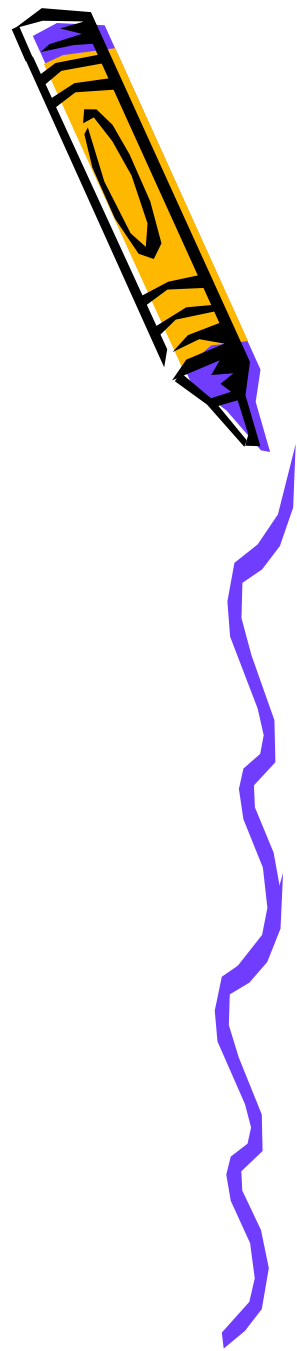
Application

- 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.



Mathematical Models

Hidden Graphs
(Reconstructed)



• Models

1) Multi-vertex model

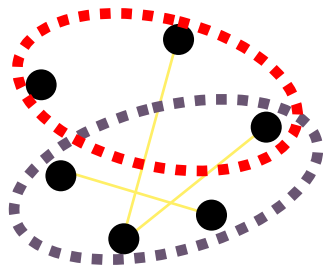
2) Quantitative multi-vertex model

3) k -vertex model

4) Quantitative k -multi-vertex model

Learning a hidden graph by edge-detecting queries:

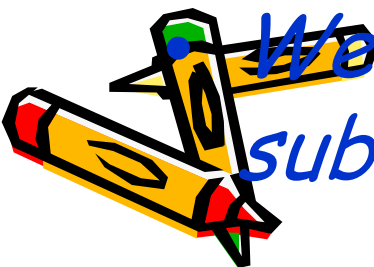
$Q_G(S)$: does S include at least one edge of G ?



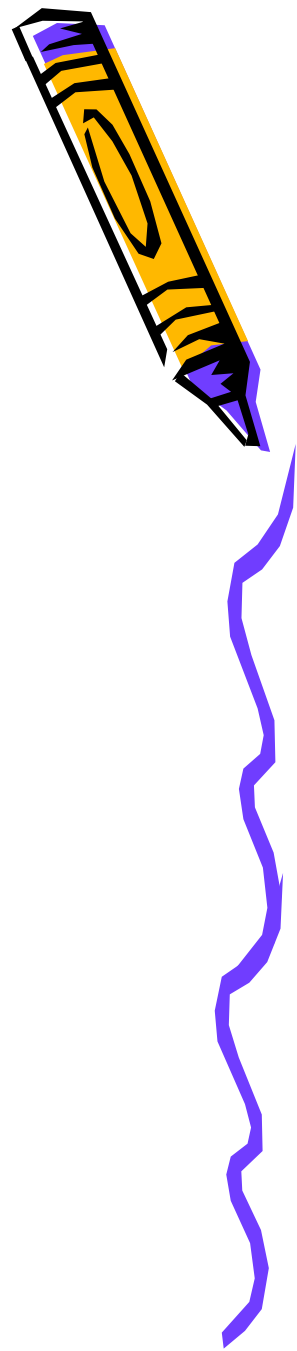


Then, ...

- We can delete the edge and start it over again to find another edge until all the edges are found.
- Clearly, if the hidden graph we are looking for is of large size compare to n , then this algorithm is not going to be a good one.



We may simply ask all the $n(n-1)/2$ 2-subsets of $[n]$.

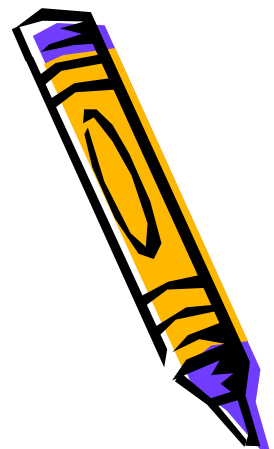


What is algorithm?

- 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.



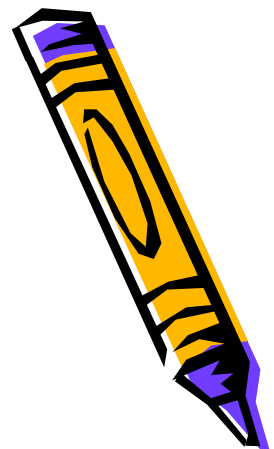
Al-Khowarizmi [Al-khOwArEz'mE]



Algorithm

- *Algorithms* play the most important role in the study of Computational Molecular Biology.
- Are there any “new” ideas which can be applied in this study? More combinatorial theory? Discrete Mathematics?



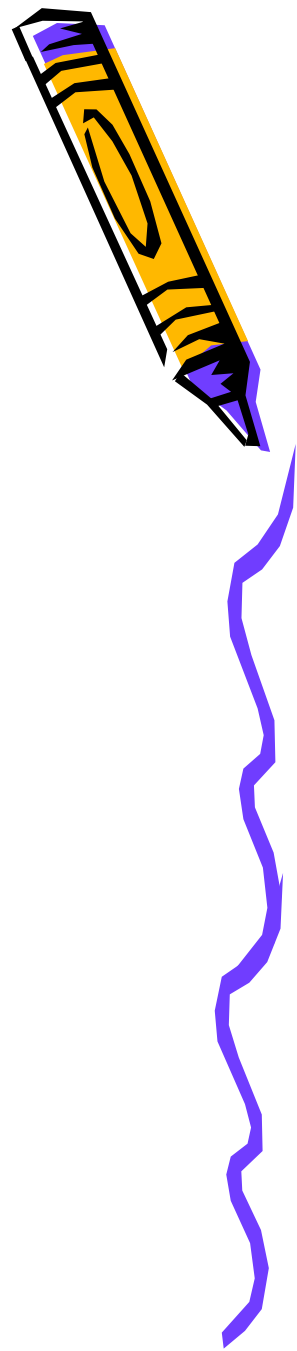


Why should we study algorithms?

- 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	$> 10^{100}$	$> 10^{100}$





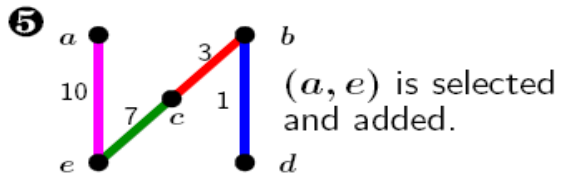
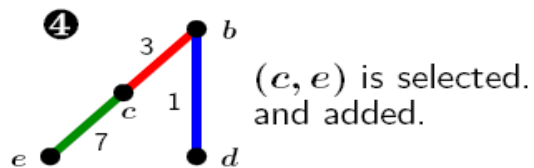
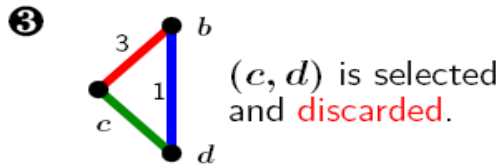
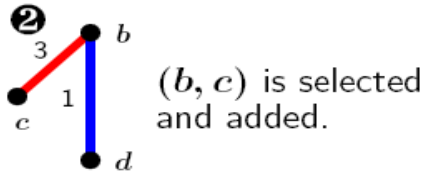
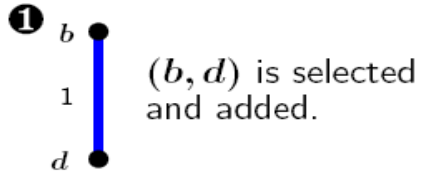
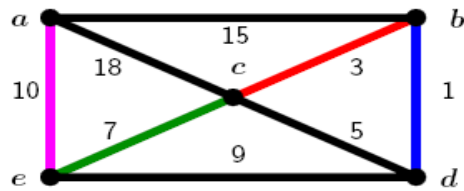
How to design a good algorithm?

- Efficient or not?
(Efficient means short time and small space.)
- Strategies of algorithms:
 1. Greedy
 2. Divide & conquer
 3. Prune & search
 4. Dynamic programming
 5. Branch and bound
 6. Approximation
 7. Heuristics

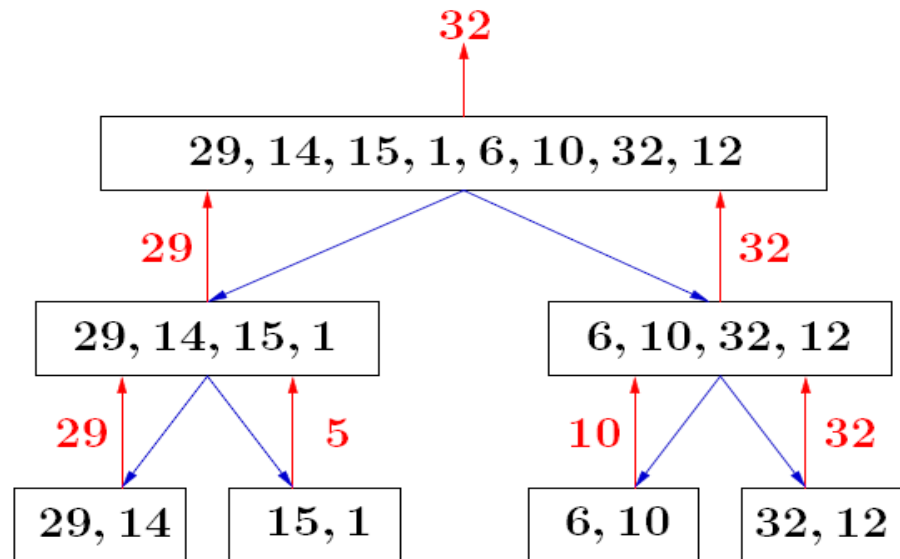


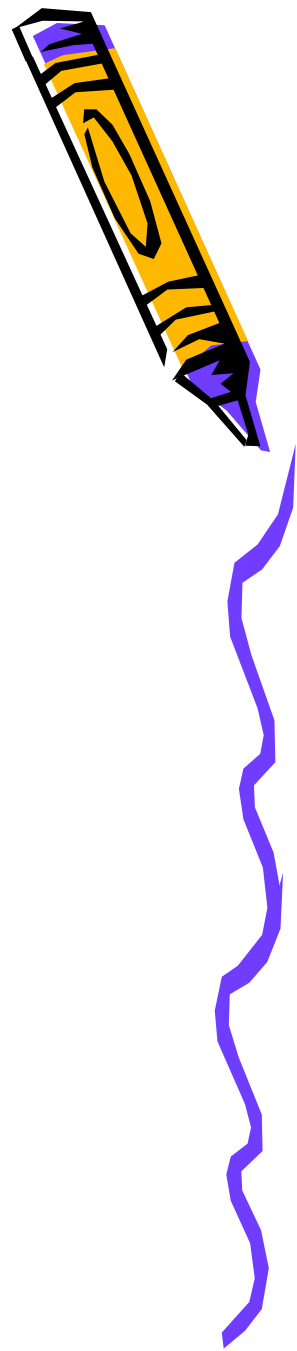


Kruskal's algorithm for MST



MaxFinding problem





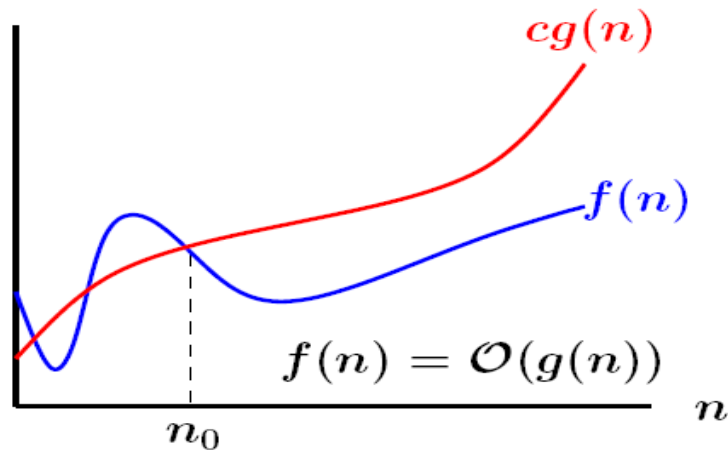
Time-complexity of an algorithm

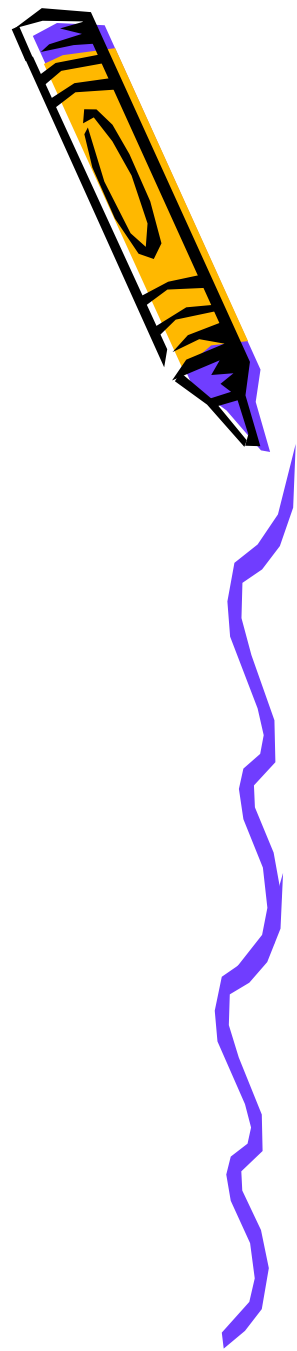
- The time of executing an algorithm is dependent on the size of the problem, denoted by n usually.
 - Most algorithms need more time to complete when n increases.
 - The time-complexity of an algorithm is usually represented by a function of n .
- Suppose it takes $n^3 + n$ steps to run an algorithm.
 - We say that the time-complexity of this algorithm is in the order of n^3 .
 - n^3 dominates n as n becomes very large.



$\mathcal{O}$ notation

$f(n) = \mathcal{O}(g(n))$ iff there exist two positive constants c and n_0 such that $f(n) \leq cg(n)$ for all $n \geq n_0$.





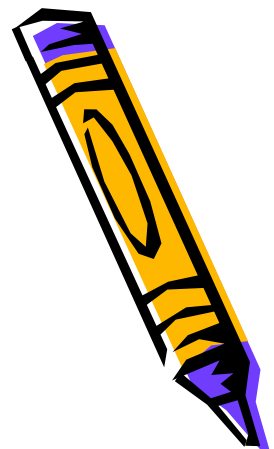
Have a try...

- Suppose that it takes $n^3 + n$ steps to run an algorithm $\mathcal{A}$.
- Then we can say that the time-complexity of $\mathcal{A}$ is $\mathcal{O}(n^3)$. (why?)
- Prove $f(n) = \mathcal{O}(n^3)$ if let $f(n) = n^3 + n$.

$$\begin{aligned} f(n) &= n^3 + n \\ &= \left(1 + \frac{1}{n^2}\right)n^3 \\ &\leq 2n^3 \quad \text{for } n \geq 1 \end{aligned}$$

$\therefore$ we have $g(n) = n^3$, $c = 2$ and $n_0 = 1$.





Have other tries...

• Let $f(n) = \frac{1}{2}n^2 - 3n$. Then

1. $f(n) = \mathcal{O}(n^2)$ (✓)

2. $f(n) = \mathcal{O}(n^3)$ (✓)

3. $f(n) = \mathcal{O}(n)$ (✗)

4. $f(n) = \mathcal{O}(1)$ (✗)

• Let $f(n) = 2^{2005}n^2 - 3n$. Then

1. $f(n) = \mathcal{O}(n^2)$ (?)

2. $f(n) = \mathcal{O}(n^3)$ (?)

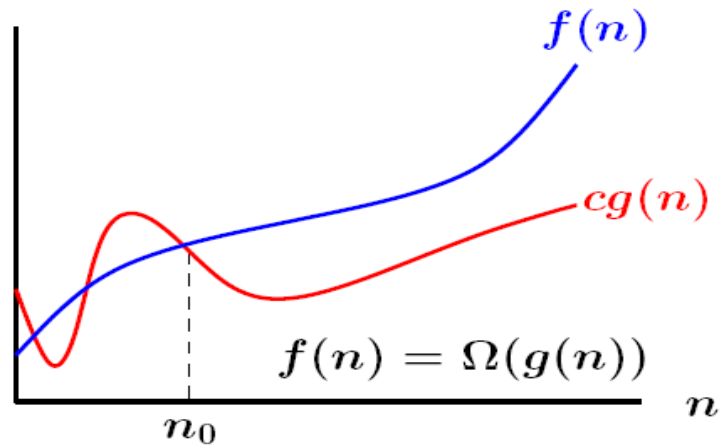
3. $f(n) = \mathcal{O}(n)$ (?)

4. $f(n) = \mathcal{O}(1)$ (?)



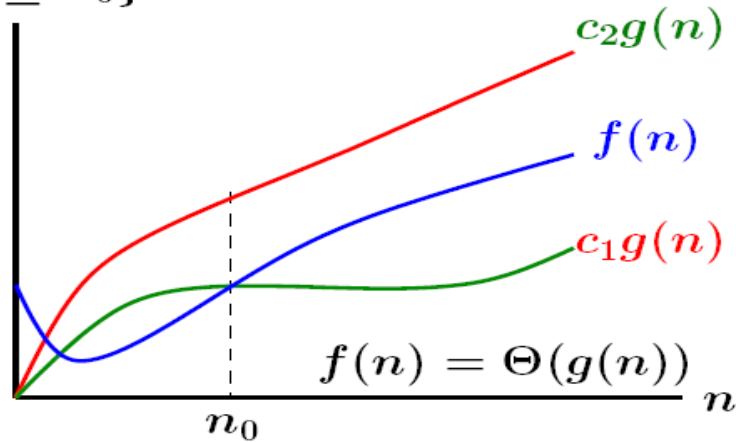
Ω notation

$f(n) = \Omega(g(n))$ iff there exist two positive constants c and n_0 such that $f(n) \geq cg(n)$ for all $n \geq n_0$.



Θ notation

$f(n) = \Theta(g(n))$ iff there exist positive constants c_1, c_2 and n_0 such that $c_1g(n) \leq f(n) \leq c_2g(n)$ for all $n \geq n_0$.





Polynomial/Exponential algorithms

- **Polynomial algorithm:**
whose complexity is bound by $\mathcal{O}(n^k)$,
where n is the input size and k is a constant
- **Exponential algorithm:**
whose complexity is bound by $\mathcal{O}(k^n)$
- **Example:** For MST problem,
 - Prim and Kruskal's methods: polynomial
 - Brute force method: exponential
- Polynomial algorithms are better than exponential ones.



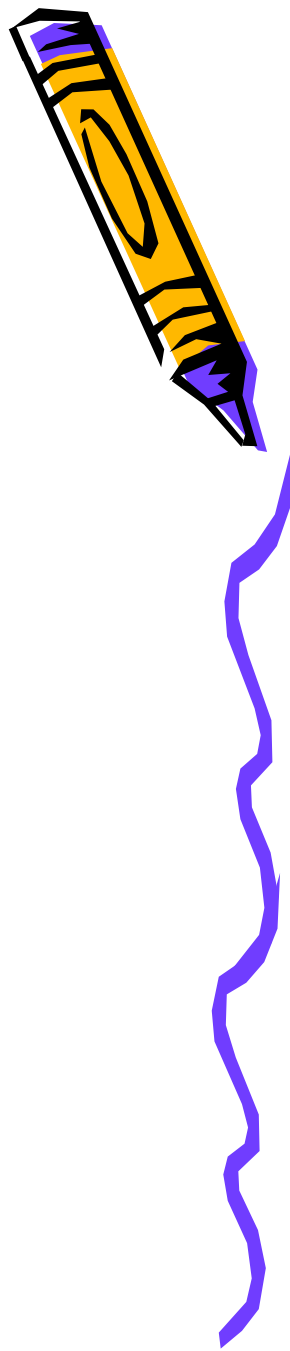
The Satisfiability Problem (SAT)

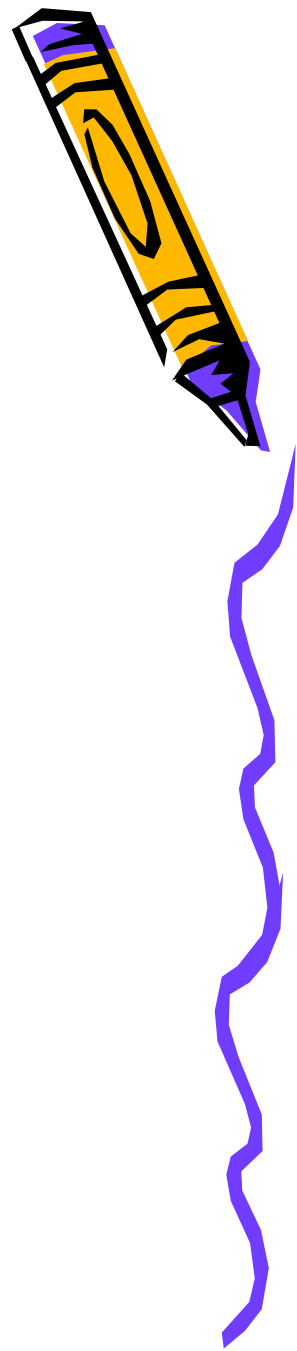
- Given a Boolean formula, determine whether this formula is satisfiable or not.
- Consider the following formula:

$$\begin{aligned} & (x_1 \vee x_2 \vee x_3) \\ & \wedge (\neg x_1) \\ & \wedge (\neg x_2) \end{aligned}$$

The following **assignment** makes the formula true.

$$\begin{aligned} x_1 & \leftarrow F \\ x_2 & \leftarrow F \\ x_3 & \leftarrow T \end{aligned}$$

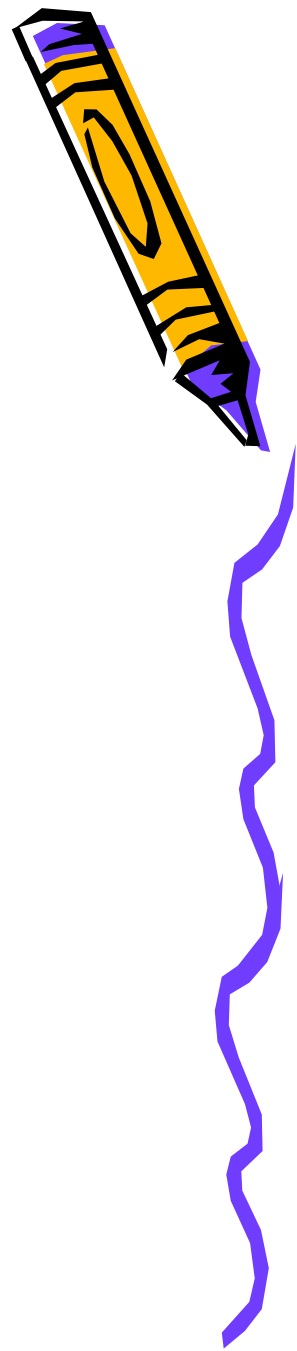




Time-complexity of SAT problem

- If there are n variables, then there are 2^n possible assignments for the SAT problem.
- Up to now, for the best available algorithms for the SAT problem, they cost exponential time in worst cases.
- **Is there any possibility that the SAT problem can be solved in polynomial time?**
- By the theory of NP-completeness, if the SAT problem can be solved in polynomial time, then all NP problems can be solved in polynomial time.





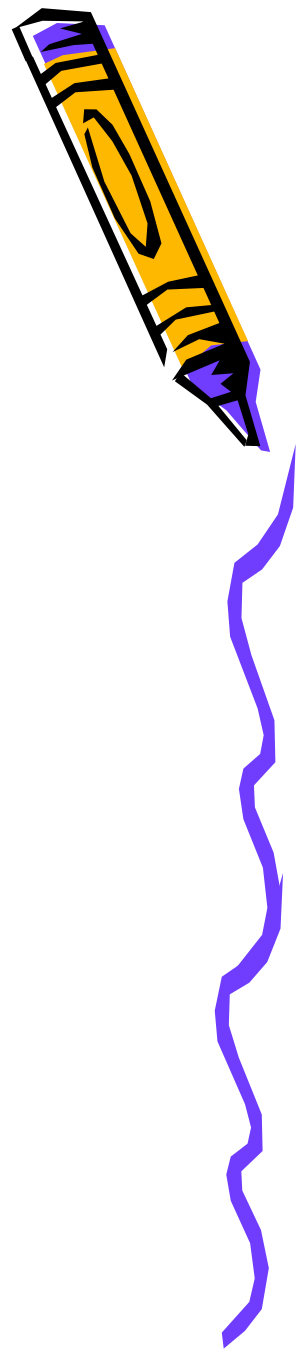
Nondeterministic algorithm

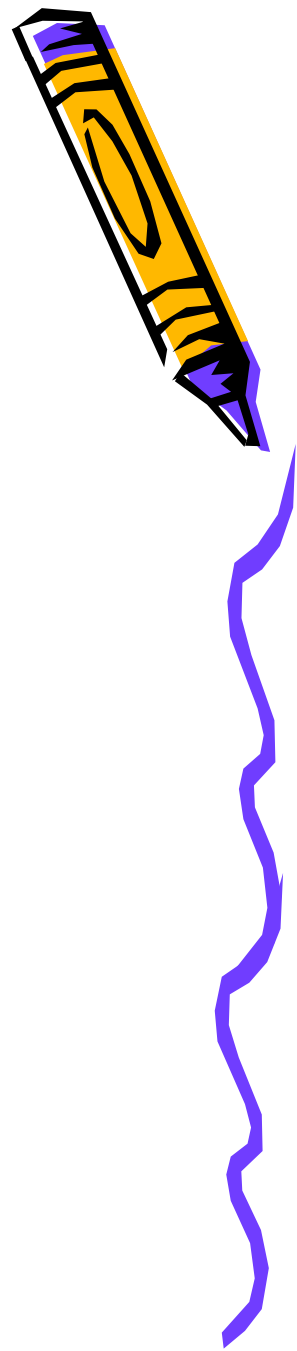
- We may consider a nondeterministic algorithm as a algorithm consisting of two phases **guessing** and **checking**.
- Given two numbers $x(1) = 7$ and $x(2) \neq 7$, determine if there is a number which equals 7.
Guessing: $i = \text{choice}(1, 2)$;
Checking: if $x(i) = 7$ then SUCCESS
 else FAILURE.
- **Nondeterministic polynomial algorithm:**
a nondeterministic algorithm whose **checking stage**
can be done in polynomial time



P and NP problems

- **P problem**: a decision problem which can be solved by a polynomial algorithm, such as the MST decision problem and the longest common subsequence decision problem
- **NP problem**: a decision problem which can be solved by a nondeterministic polynomial algorithm, such as the SAT problem and the traveling salesperson decision problem
- Every P problem must be an NP problem (i.e., $P \subseteq NP$).

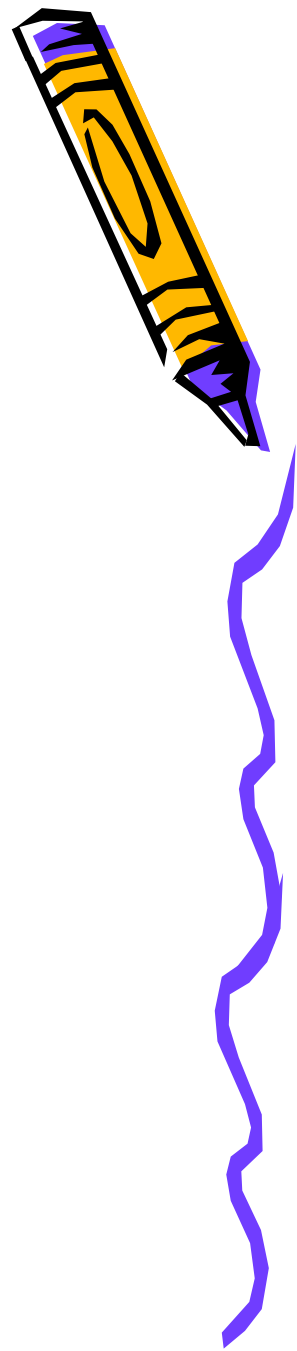




P and NP problems

- Are all problems NP problems?
 - Halting problem: given an arbitrary program with an arbitrary input data, will the program terminate or not?
 - Halting problem is not an NP problem because it is undecidable.
- Are all NP problems P problems? ($P = NP$?)
 - $P \subseteq NP$? (yes)
 - $P \supseteq NP$? (?): the SAT problem and the traveling salesperson decision problem can be solved in $O(2^n)$ and $O(n!)$ time, respectively.





NP-complete problem

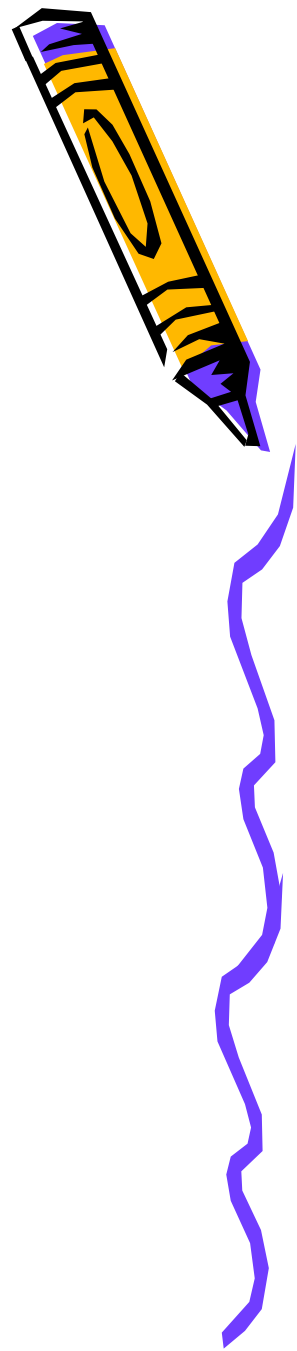
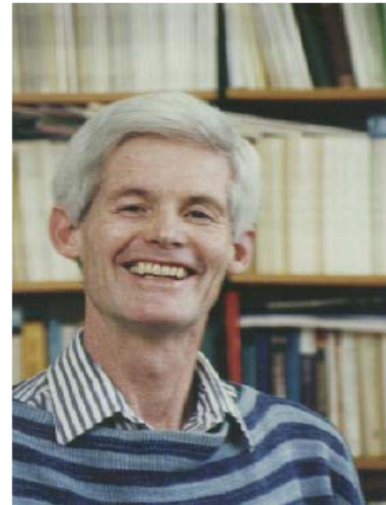
- In discussing NP problems, we shall **only discuss decision problems**.
- Problem $\mathcal{P}_1$ reduces to problem $\mathcal{P}_2$ ($\mathcal{P}_1 \propto \mathcal{P}_2$):
 - $\mathcal{P}_1$ can be solved in polynomial time by using a polynomial time algorithm solving $\mathcal{P}_2$.
 - $\mathcal{P}_2$ is more difficult than $\mathcal{P}_1$.
- A problem $\mathcal{P}$ is **NP-complete** if
 1. $\mathcal{P} \in \text{NP}$ and
 2. $\mathcal{P}$ is **NP-hard** (every NP problem reduces to $\mathcal{P}$).
- The SAT problem was the first found NP-complete problem by Cook (1971).



Stephen Cook

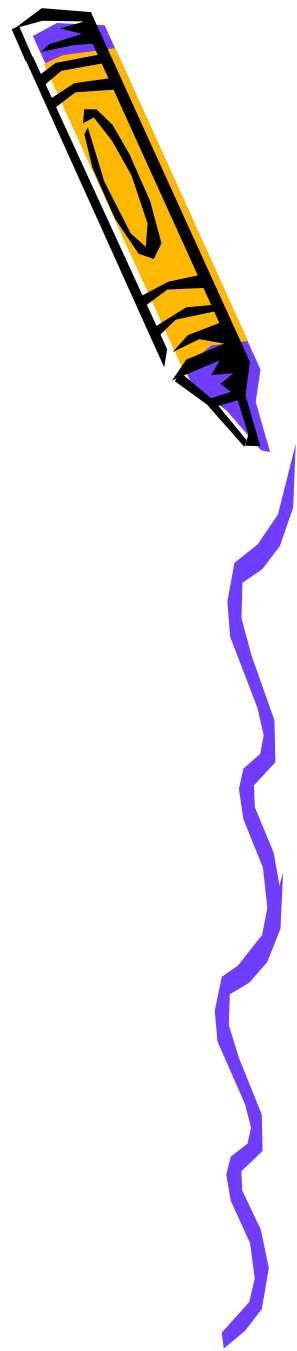
- The author of the famous paper ^a introducing the theory of NP completeness
- The 1982 recipient of the Turing award

^a The Complexity of Theorem Proving Procedures. Proceedings Third Annual ACM Symposium on Theory of Computing, May 1971, pp 151-158

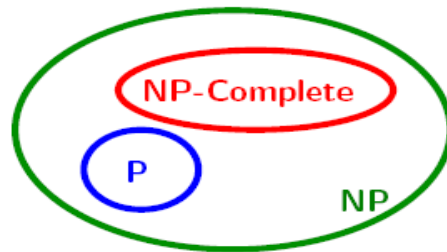


Alan Turing (1912–1954, England)

- Turing Award (accompanied by \$100,000) is given to an individual with contributions that are of lasting and major technical importance to the computer field.

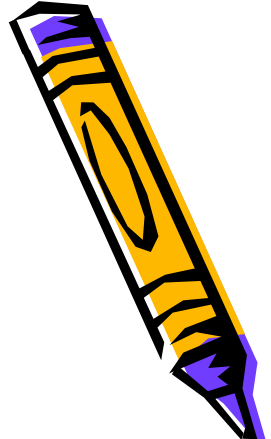


Theory of NP-Completeness



- If any NP-complete problem can be solved in polynomial time, $NP = P$.
- Up to now, no NP-complete problem has any worst case polynomial algorithm.
- There are thousands of problems proved to be NP-complete problems.
- If the decision version of an optimization problem is NP-complete, this optimization problem is called NP-hard.





The millennium prize problems

- Seven problems considered to be "important classic questions that have resisted solution over the years"
 1. P versus NP
 2. The Hodge Conjecture
 3. The Poincare' Conjecture
 4. The Riemann Hypothesis
 5. Yang-Mills Existence and Mass Gap
 6. Navier-Stokes Existence and Smoothness
 7. The Birch and Swinnerton-Dyer Conjecture
 - The first person to solve each problem above will be awarded \$1,000,000 by CMI.
- 