P vs NP Conjecture

- A 40 year old open problem in Mathematics/Theoretical Computer Science
- Has a million dollar prize (from Clay Mathematical Institute)
- It is a question about `finding solutions' and `verifying solutions'

Time complexity function	Size n					
	10	20	30	40	50	60
n	.00001 second	.00002 second	.00003 second	.00004 second	.00005 second	.00006 second
n^2	.0001 second	.0004 second	.0009 second	.0016 second	.0025 second	.0036 second
n^3	.001 second	.008 second	.027 second	.064 second	.125 second	.216 second
n^5	.1 second	3.2 seconds	24.3 seconds	1.7 minutes	5.2 minutes	13.0 minutes
2^n	.001 second	1.0 second	17.9 minutes	12.7 days	35.7 years	366 centuries
3^n	.059 second	58 minutes	6.5 years	3855 centuries	2×10^8 centuries	1.3×10^{13} centuries

Figure 1.2 Comparison of several polynomial and exponential time complexity functions.

Example

- Typical input lengths that users and programmers are interested in are approximately between 100 and 1,000,000. Consider an input length of n=100 and a polynomial algorithm whose running time is n^2. This is a typical running time for a polynomial algorithm.) The number of steps that it will require, for n=100, is 100^2=10000. A typical CPU will be able to do approximately 10^9 operations per second (this is extremely simplified). So this algorithm will finish on the order of (10000 $\div 10^9$) = .00001 seconds. A running time of .00001 seconds is reasonable.

Size of Largest Problem Instance
Solvable in 1 Hour

Time complexity function	With present computer	With computer 100 times faster	With computer 1000 times faster
n	N_1	$100\ N_1$	$1000\ N_1$
n^2	N_2	$10\ N_2$	$31.6\ N_2$
n^3	N_3	$4.64\ N_3$	$10\ N_3$
n^5	N_4	$2.5\ N_4$	$3.98\ N_4$
2^n	N_5	$N_5 + 6.64$	$N_5 + 9.97$
3^n	N_6	$N_6 + 4.19$	$N_6 + 6.29$

Figure 1.3 Effect of improved technology on several polynomial and exponential time algorithms.

Efficient Algorithms

- Do all problem have efficient solution.
- No
- For a large class of natural problems, no efficient solution exists.

Generating Vs Checking

- Factorise a large number which is product of 2 primes.
- Stduent: Given N, find p,q such that pq=N. generate a solution.
- Teacher: verify pq=N. checks.

Checking algorithm

- Takes an input instance I and a solution yes or no.
- Boolean satisfiability:
- X,y,z variables, not x, y, z
- Clause – Disjunction-formula of the form (x||y||!z)
- Formula: Conjunction C1 and C2 and C3 and C4

- Example:
- (x||y||z)&(x||!y)&(y||!z)&(!x||!y||!z)
- Solution: x = true, y = true, z= false

(x||y||z)&(x||!y)&(y||!z)&(z||!x)&(!x||!y||!z)

Solution: No assignment.

Solution

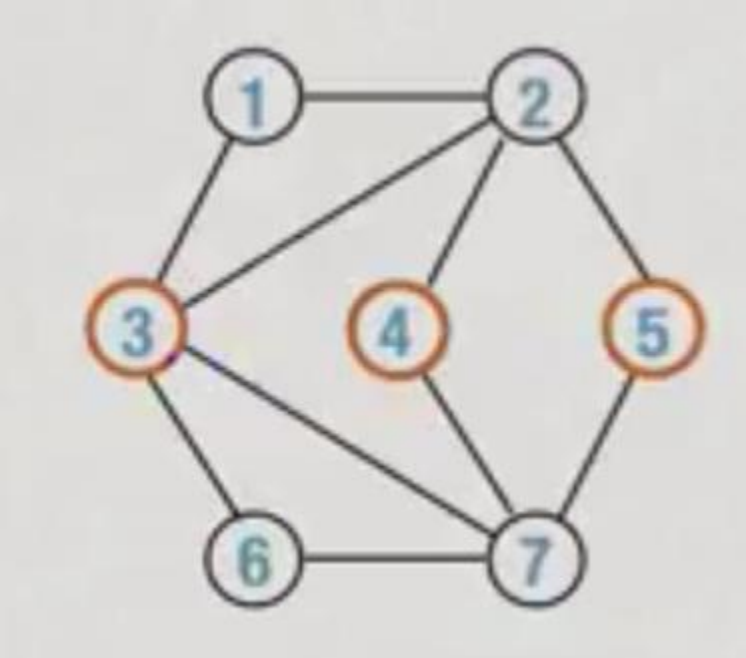
- Bruteforce technique: 2^N
- No better solution.
- Algorithm to check the solution in polynomial time.
- Generting algorithm is 2^N

TSP

- Generating algorithm:
- O(n!)
- $O(n^2 2^n)$- DP
- G= (V,E), tour x,y,z,…..x with minimal cost.

- Checking algorithm:
- Solution s is given:
- Verify s is cycle.
- Compute its cost.
- How to check is it minimal? Upper bound.

Independent set



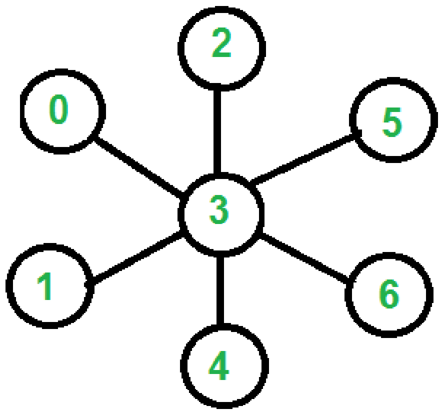
Independent set

- U,V are independent if there is no edge (u,v).
- Find the largest independent set in the graph.
- (1,5), (1,5,7) is not because 5,7 is connected.
- (3,4,5) is independent set of size 3.

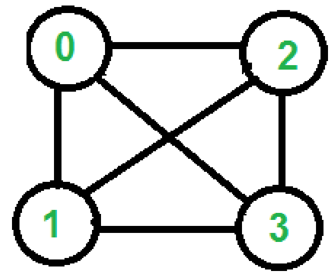
Vertex cover

- A vertex cover of an undirected graph is a subset of its vertices such that for every edge (u, v) of the graph, either ‘u’ or ‘v’ is in vertex cover. Although the name is Vertex Cover, the set covers all edges of the given graph.***Given an undirected graph, the vertex cover problem is to find minimum size vertex cover***.

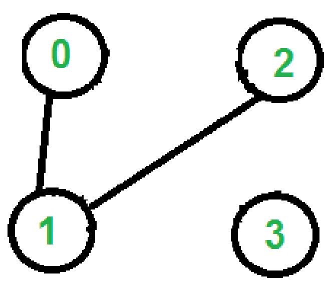
Example



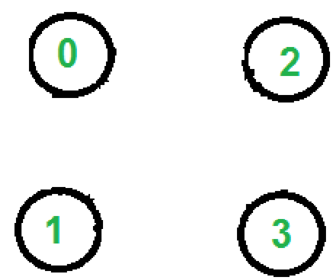
Minimum Vertex Cover is {3}



Minimum Vertex Cover is {0, 1, 2} or {0, 1, 3} or {1, 2, 3}

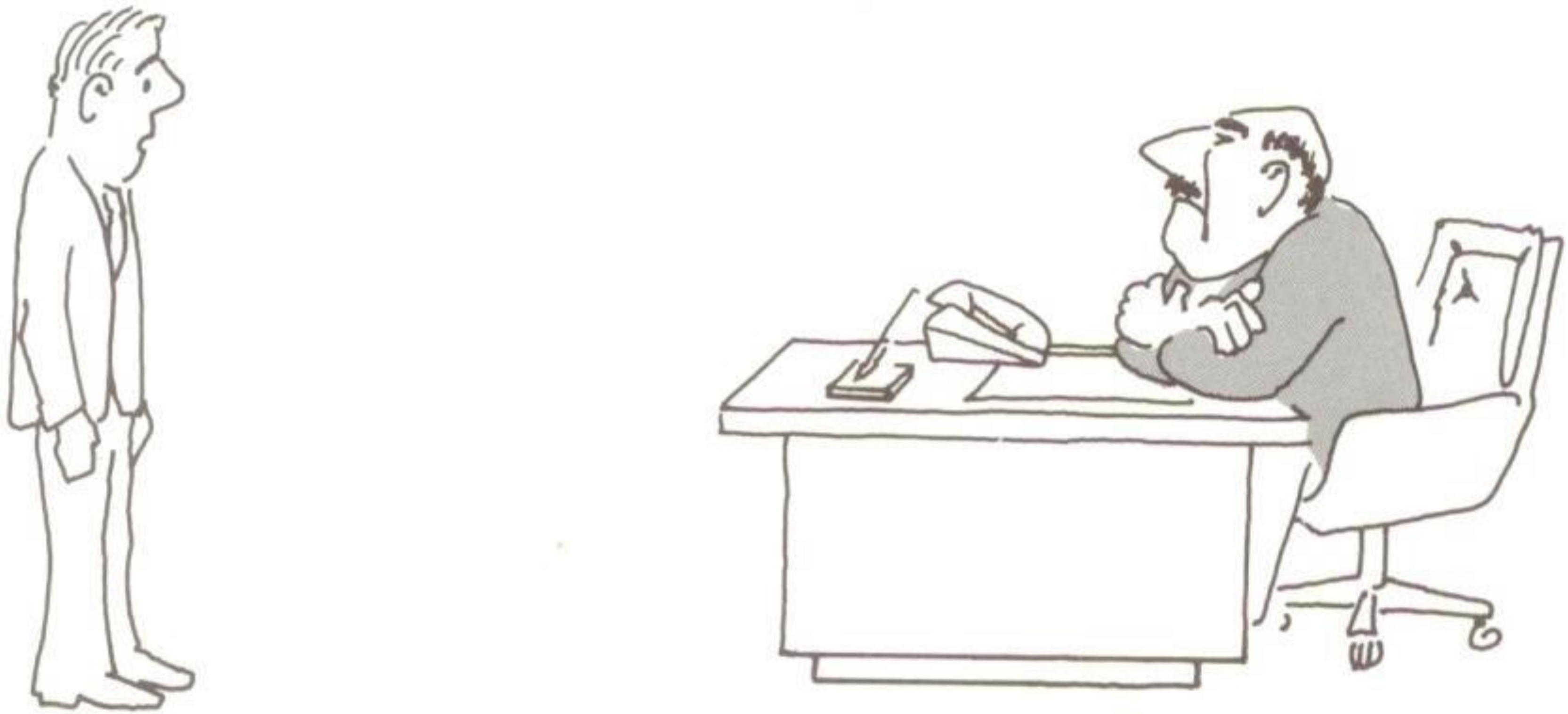


Minimum Vertex Cover is {1}



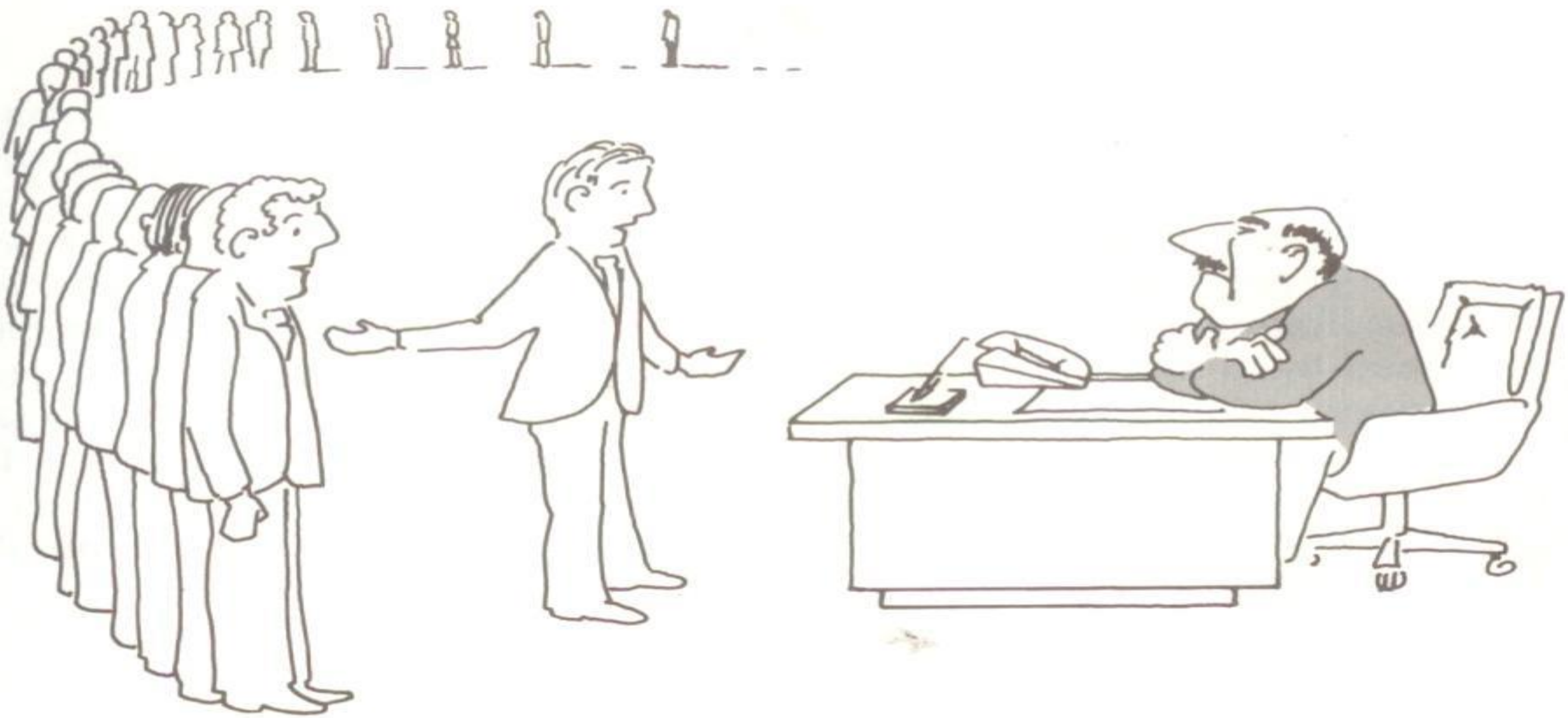
Minimum Vertex Cover is empty {}

- Checking algorithm: is there a vertex cover of size k?
- Vertex cover of size k= complement of independent set of size n-k.



“I can’t find an efficient algorithm, I guess I’m just too dumb.”

"I can't find an efficient algorithm, because no such algorithm is possible!"



“I can’t find an efficient algorithm, but neither can all these famous people.”