Almost all optimization problems can be converted to its corresponding decision problems.

Q:

An efficient algorithm is there for an optimization problem.

What about the efficiency of its corresponding decision problem?

if optimization can be solved efficiently then decision version can also be solved efficiently.

contrapositive: Decision version not solved efficiently (hard to solve)

then optimization version not solved efficiently (hard to solve)

yes input / instance
no input / instance

Ex^m Is $a+b=c$ for $a, b, c \in \mathbb{N}$?

(a, b, c)

$(1, 2, 3)$ $1+2=3$
yes input

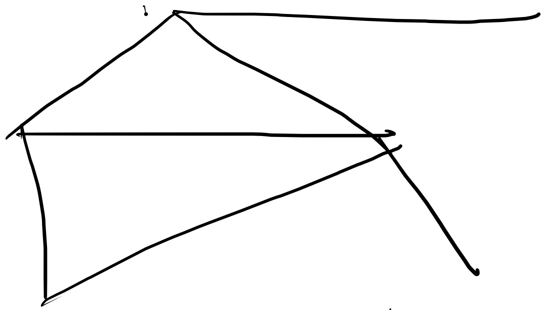
$(5\ 6\ 9)$ $5+6=9$ X
no input.

Def^n An instance of a decision problem is called an yes instance if the answer to the instance is yes.

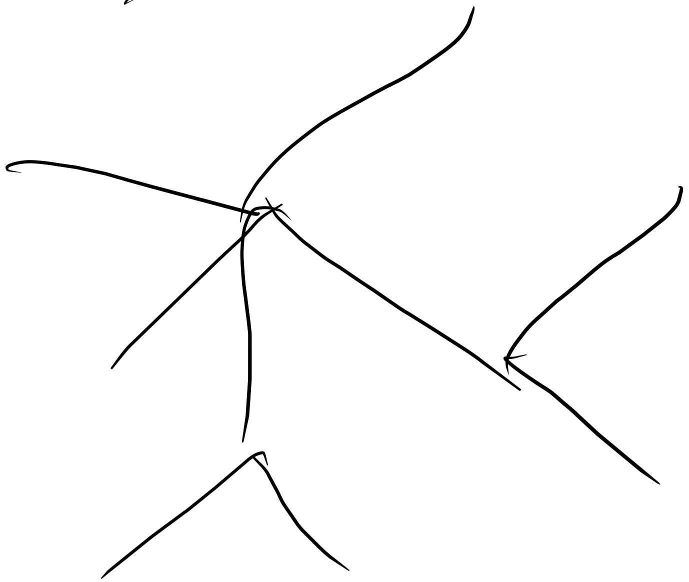
otherwise if the answer is no it is a no instance.

Ex^{m} Does an undirected graph G contains a cycle?

F



Yes instance.



No instance

complementary problems

L be a decision problem then $\overline{L}$ is the decision problem such that yes-input of $\overline{L}$ are exactly the no-input of L.

Exm: composite: Is a given positive integer n is composite?

$$n = ab \quad \text{for } 2 \le a \le b < n$$

Prime: Is a given number n is prime?

$\overline{\text{composite}}$ = prime.

$\overline{\text{prime}}$ = composite.

Property:

$$\overline{\overline{L}} = L$$

Polynomial-time algorithms

If an algorithm runs in $O(n^k)$ time

where k is a constant independent of n.

n: input size

Ex^m matrix multiplication : $O(n^3)$

Remark: n or n^c or cn as input size for c: constant

$O(n^k)$ Now $O((n^c)^k) = O(n^{ck})$

Ex^m MST : $O(m \log n)$

Non-polynomial time algorithm.

running time is not $O(n^K)$

Exm Prime: Decide whether a positive integer I is a prime or not?

Algo:

for $i = 2$ to $\sqrt{I}$

check whether i devides I or not.

Input: I

Input size $n = \log I$

$\Rightarrow I = 2^{n}$

running time: $O(I^{1/2})$

Is this polynomial?

running time: $O((2^{n})^{1/2})$

Polynomial v non-polynomial

running time of an algorithm is $O(2^n)$

Let $n = 100$

A computer performs 10^{12} operations per second.

~~on~~ Time taken: $\dfrac{2^{100}}{10^{12}} \simeq 10^{18.1}$ seconds.

$\simeq 4 \cdot 10^{10}$ years.

Remark: for polynomial time algorithm large exponent is also impractical.

Polynomial-time algorithm $\Longrightarrow$ tractable.

The class P

class of all decision problems that are solvable in polynomial time.

Q: How to prove a decision problem is in P?

A: by designing a polynomial time algorithm.

Q: How to prove it is not in P?

No polynomial time algorithm exists.