April 28

The question: are there problems that can't be solved efficiently no matter how hard we work?

Computational Complexity / NP-C. NP-hard

Efficiently = polynomial time solution/algorithm

① Polynomial Time Reduction

② P. NP, NP-hard NP-C

③ How to show a problem is NP-hard by reduction

Problem: 2-partition.

Given a set S of n non-negative integral numbers.

$S = \{s_1, s_2, \ldots, s_n\}$

Is there a subset S_1 of S such that the sum of all the integers belong to S_1 is exactly $\frac{1}{2}$ of the sum of all the numbers in S?

Find S_1 such that

$$\sum_{s' \in S_1} s' = \frac{1}{2} \sum_{s \in S} s$$

Example: $S = \{1, 2, 3, 4\}$

Answer: Yes, $S_1 = \{1, 4\}$

Solution:

(1) For each $s_j \in S$, we will introduce an item t_j whose value (c_j) is s_j and whose size (k_j) is also s_j.

Let the Knapsack size $K = \frac{1}{2} \sum_{s_j \in S} s_j$

(2) The Knapsack Problem

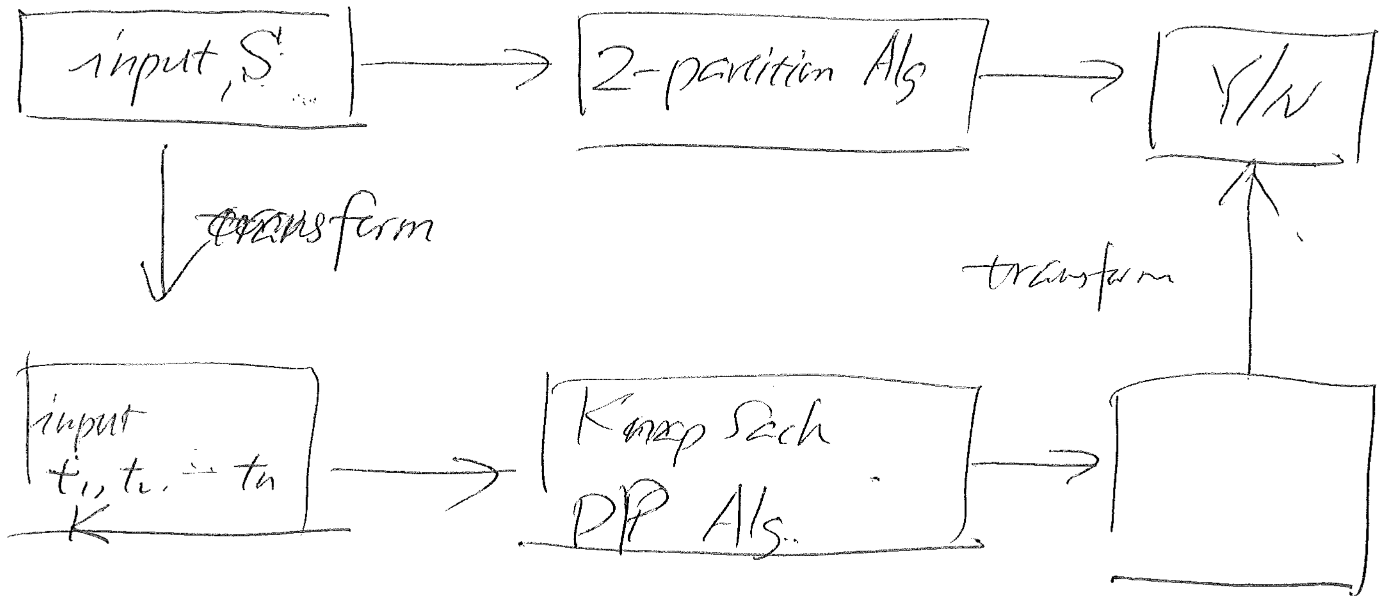
items: $t_1, t_2, \ldots, t_n$

K

Find a subset of items to fit in the Knapsack while maximizing the value.

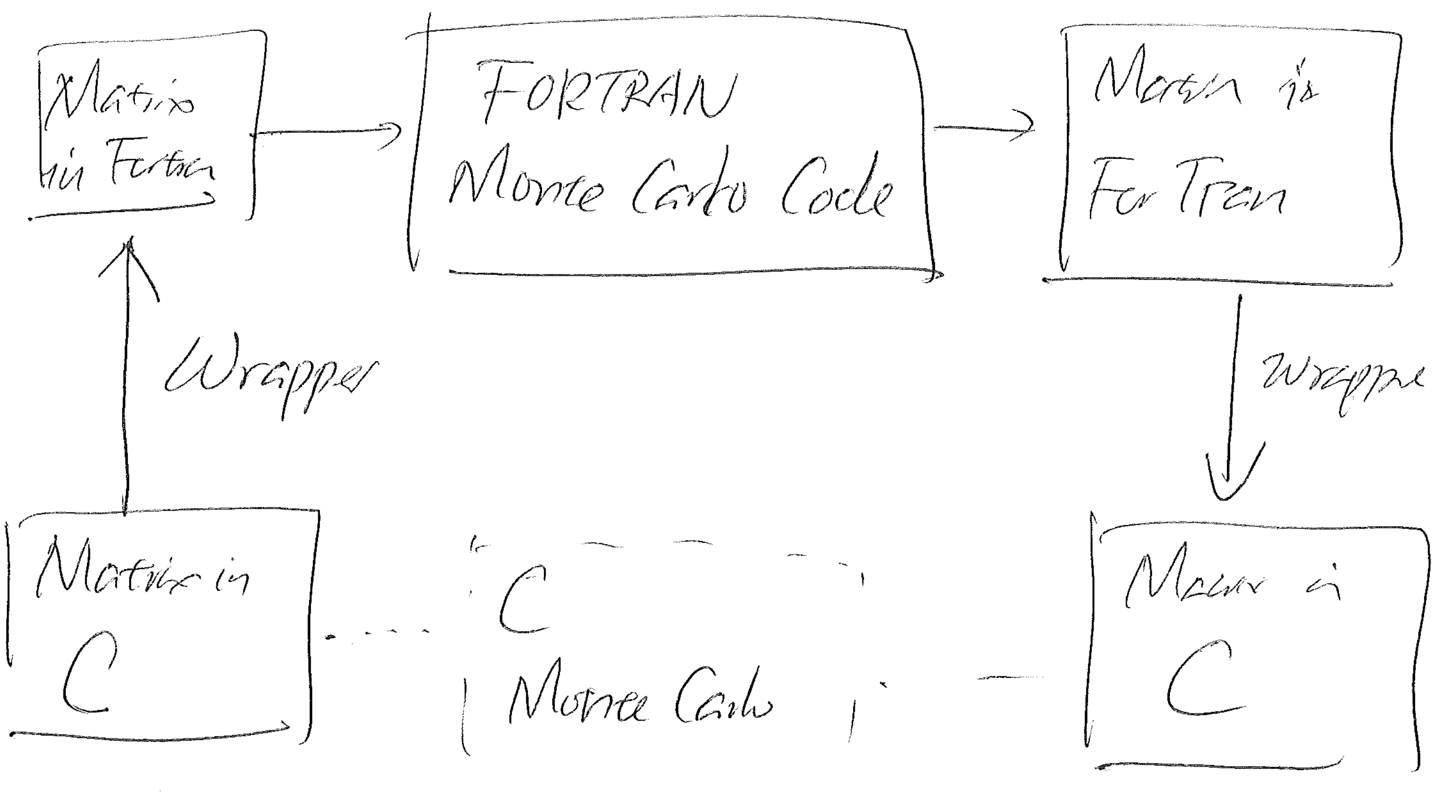
(3) If the optimal solution has value K, answer Yes to 2-partition

Reduction:



Example:

① Wrapper Software



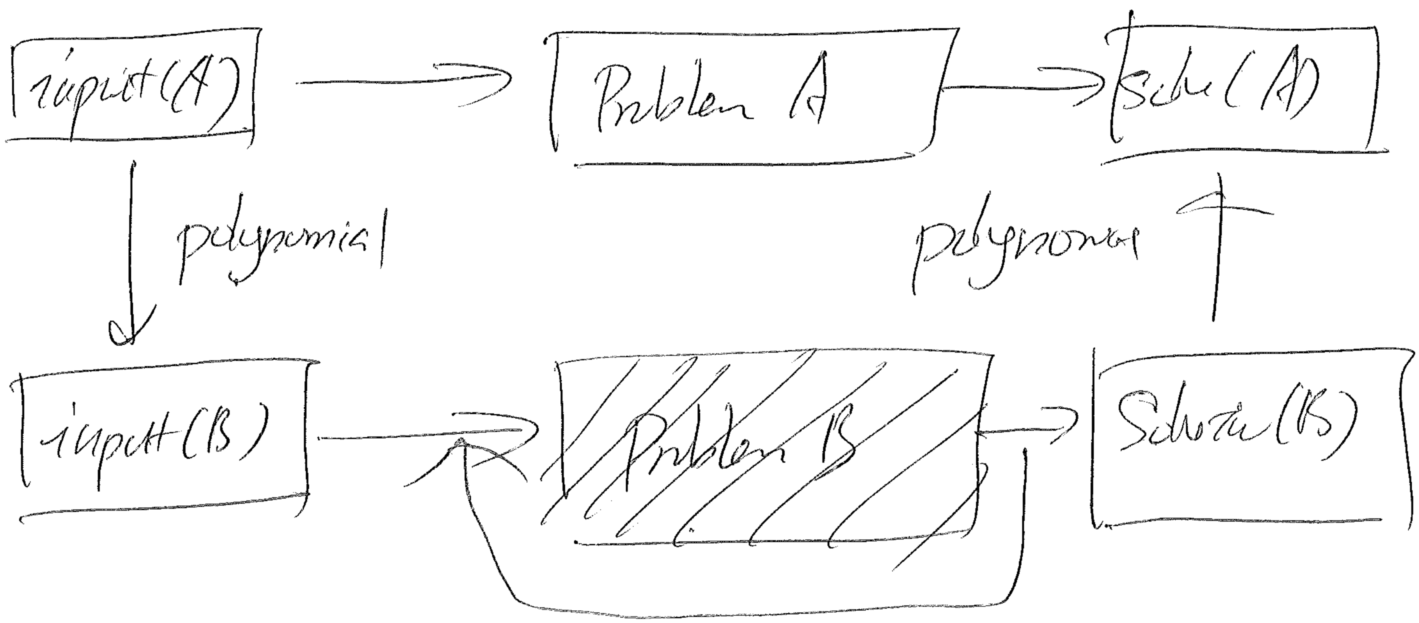
Polynomial Time Reduction

Let A, B be two ~~algs~~ algorithmic problems

A: input $(A) \longrightarrow$ solution (A)

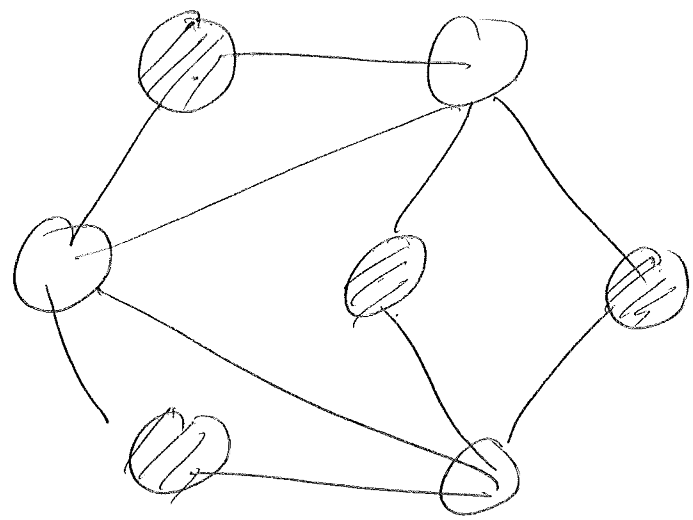
B: input $(B) \longrightarrow$ solution (B)

We say that A is polynomial time reducible to B if arbitrary input instance of A can be solved using a polynomial number of discrete steps for conversion of ~~the~~ input (A) to input (B) and solution (B) to solution (A) plus a polynomial number of calls to the blackbox for solving B.



Example:

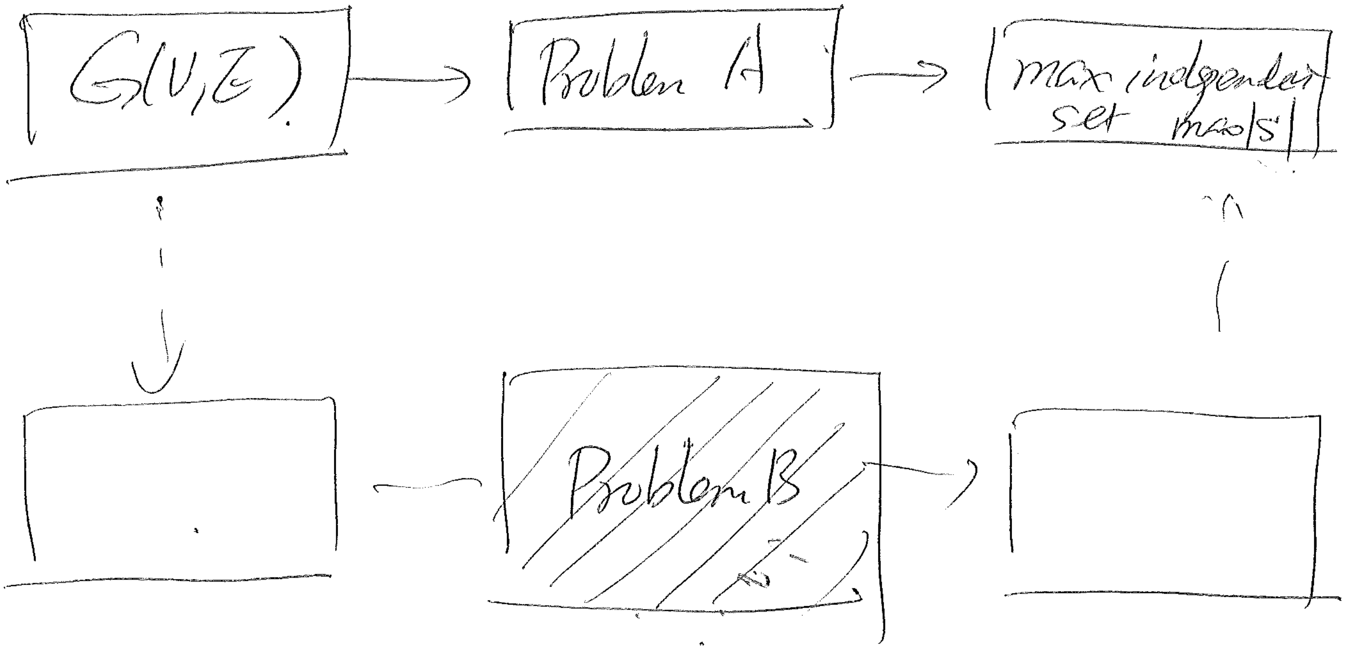
independent set: Given an undirected graph $G(V,E)$ an independent set is a subset S' of vertices, such that no two vertices in S' are adjacent (joined by an edge in E)



Problem A: maximum independent set
maximize $|S'|$.

Problem B: Given $G(V,E)$ and an integer $k \leq |V|$
Is there an independent set of size k?

Can you reduce A to B in polynomial time?



For k=1 to |V|

if Decision/Problem B (G, k) = Yes

k++

else

return (k-1)

Observe that if we can solve B, then we can solve A

B is <u>at least as hard</u> as A

Running time of the detour

$$T(A) = \text{polynomial} + \text{polynomial} \cdot T(B)$$

$$T(A) = P(n) + Q(n) \cdot T(B)$$

Observation 1: if B can be solved in polynomial time, then A can be solved in polynomial time also

Observation 2: If A can't be solved in polynomial time, ~~then~~ then B can't be solved in polynomial time either

$$T^*(A) > \text{polynomial}$$

$$T(A) \geq T^*(A) > \text{polynomial}$$

$$P(n) + Q(n) \cdot T(B) > \text{polynomial}$$

$$T(B) > \text{polynomial}$$

Observe that if your problem is B, to show that B is NOT polynomial time solvable, it suffices to find a KNOWN problem A NOT polynomial time solvable and reduce A to B.

When we can reduce A to B in polynomial time we write $A \leq_p B$

The Class of P:

Deterministic Polynomial Time Solveble:

include all problems thate can be solved in polynomial deterministically. $O(n^K)$

All the problems — Knapsack are in P.

The Class of NP:

Non-deterministic Polynomial Time Solvable.

We say a computational problem is NP if it has a non-determinisic polynomial time algorithm., i.e., given somethig we can verify if this soln is a solution or not in polynomial time. (polynomial time verifiable)

Ex1 Sorting $\in P$.

Sorting $\in NP$.

input: 4, 5, 6, 1, 3, 7

$\boxed{4, 6, 1, 3, 7, 6, 5}$

Ex 2 Decision version of the Independent Set.

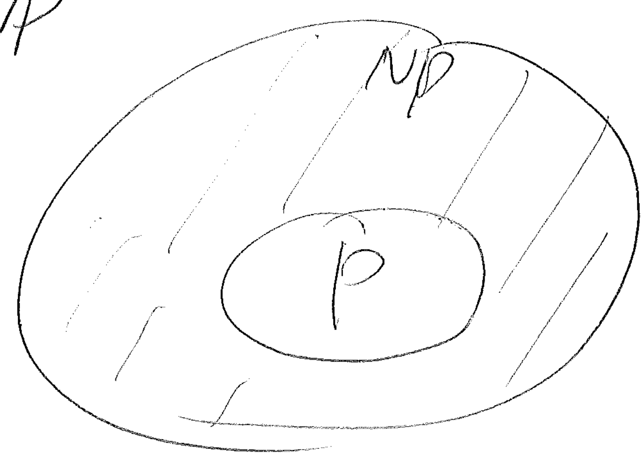
Given $G(V,E)$ $k \leq |V|$, is there an independent set of size k?

Is the problem in NP? Yes.

$P \subseteq NP$

Generally, we believe

$P \neq NP$, $P \subset NP$



What are the hardest problems in NP?

NP-hard.

Def A problem is NP-hard if it is <u>at least as hard</u> as every problem in NP.

A problem A is NP-hard if we can reduce every NP problem to A