CS F364
Design & Analysis of Algorithms

Complexity – Optimization Problems

Intractable Optimization Problems

- Complexity Classes and Reduction
- Hardness of Optimization Problems
    - Examples
- Pseudo-polynomial time algorithms
- Strongly NP-hard problems

4/30/2015 Sundar B. CSIS, BITS, Pilani

RECALL: REDUCTIONS

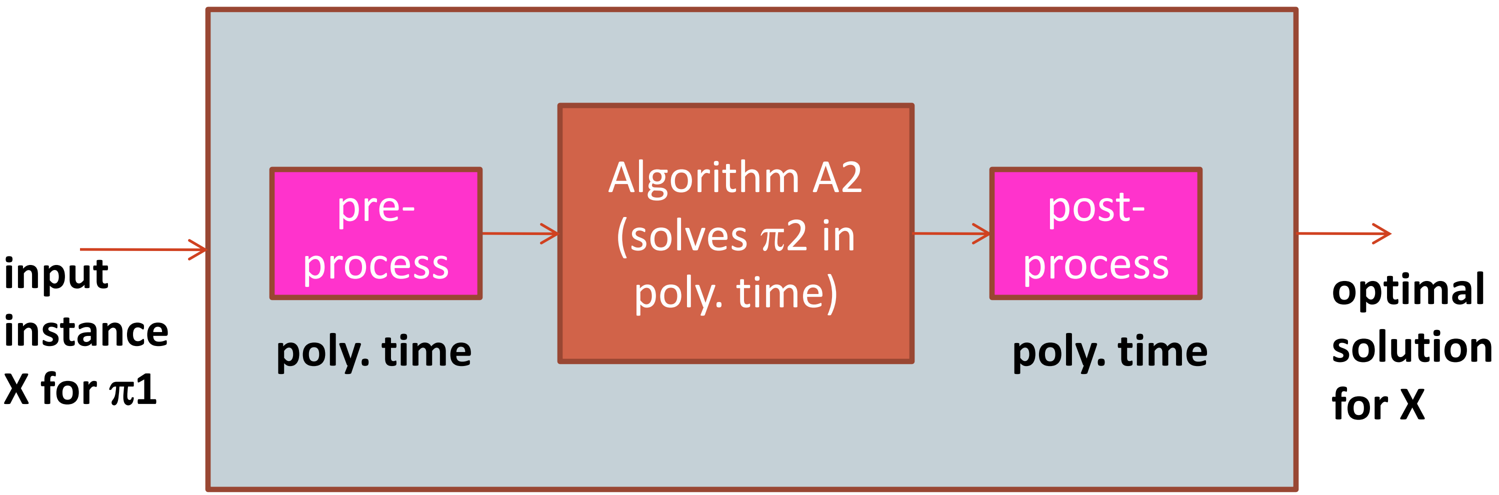
- Recall:
  - We say π_1 ***(polynomially) reduces to*** π_2
    - if there is a polynomial-time computable function $f : \mathbf{I}(\pi_1) \rightarrow \mathbf{I}(\pi_2)$ such that
    - for every $x \in \mathbf{I}(\pi_1)$,
      - $\pi_1(x) = 1$ if and only if $\pi_2(f(x)) = 1$
- We use $\pi_1 \precsim \pi_2$
  - to denote that π_1 ***(polynomially) reduces to*** π_2

RECALL: REDUCTIONS AND ORACLES

- Another way of understanding $\pi_1 \precsim \pi_2$
  - If there is an algorithm A_2 to (efficiently) solve π_2 then one can use that construct an (efficient) algorithm A_1 to solve π_1
- Alternatively,
  - If we can use a poly-time algorithm A_2 that solves π_2 to construct a poly-time algorithm A_1 to solve π_1 , then $\pi_1 \precsim \pi_2$
- Question:
  - Can we quantify "use"?
  - Karp Reduction: Single query to Oracle
  - Turing Reduction: Polynomial Number of queries to Oracle

REDUCTIONS AND ORACLES [2]

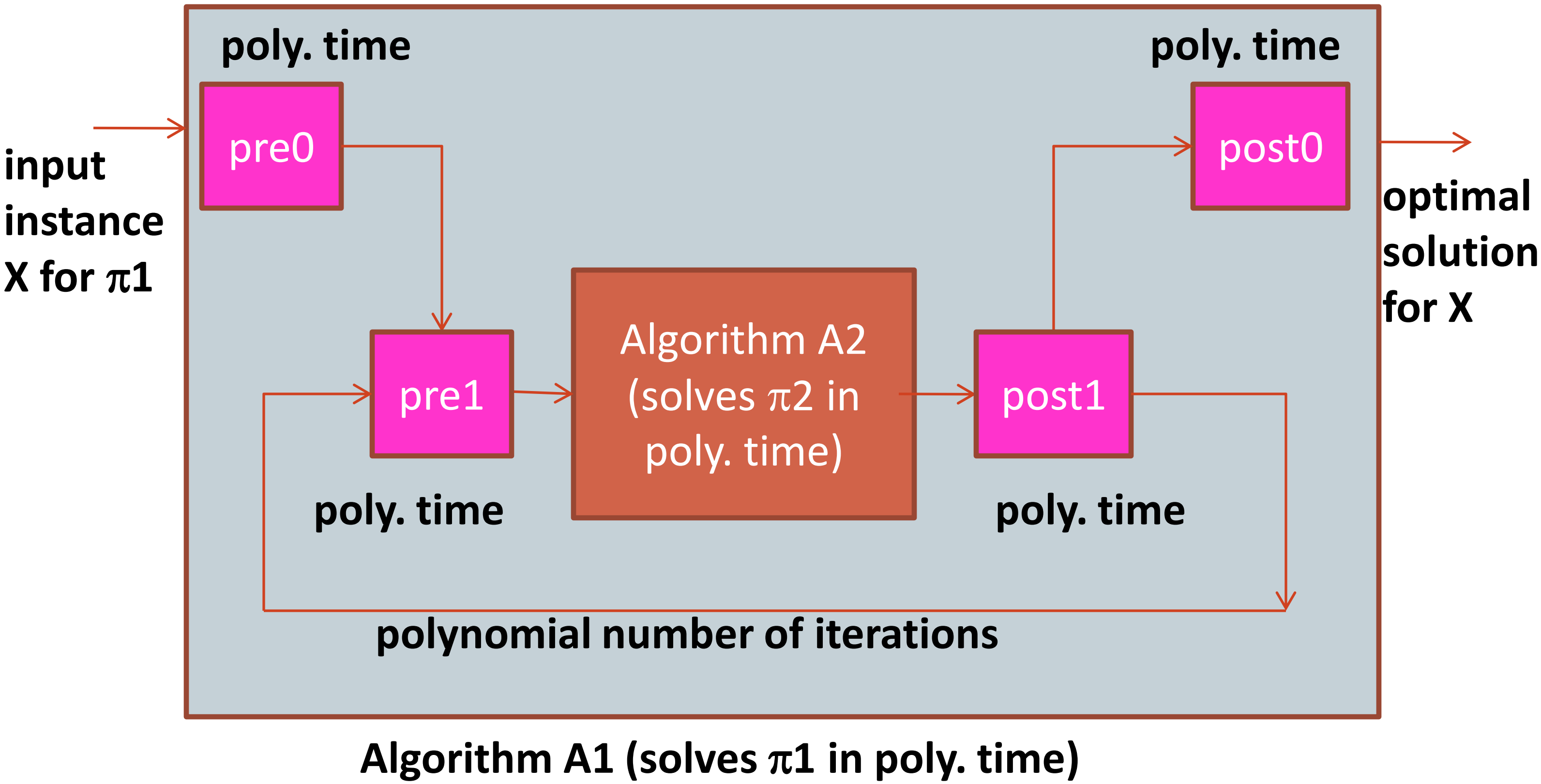
$\pi 1 \precsim \pi 2$: Karp reduction



Algorithm A1 (solves π1 in poly. time)

Reductions and Oracles [3]

$\pi1 \precsim \pi2$: Turing reduction



Hardness of Optimization problems

- An optimization problem π in $\mathbb{NPO}$ is said to be $\mathbb{NP}$-hard if for all problems π' in $\mathbb{NP}$:
  - $\pi' \precsim \pi$
- Note on reduction:
  - We will typically assume Turing reductions as a Karp reduction is a special case of a Turing reduction.

OPTIMIZATION PROBLEMS - FORMULATIONS

- An optimization problem **π = <I, SOL, m, goal>** can be formulated in different ways:
  - **Construction Version (π_c)** :
    - Given an input instance x, find the ***optimal solution* OPT(x)**
  - **Evaluation Version (π_e)** :
    - Given an input instance **x**, find ***the optimal measure*** i.e. measure of the optimal solution **$m^*(x) = m(x, OPT(x))$**
  - **Decision Version (π_d) :**
    - Given an input instance **x**, and a threshold value **k** decide whether ***the optimal measure is bounded by k***
      - i.e. Is **$m^*(x) <= k$** ? (if **goal** is *min*)
      - Is **$m^*(x) >= k$** ? (if **goal** is *max*)

OPTIMIZATION PROBLEMS – FORMULATIONS [2]

- Example: TSP
  - TSP_c :
    - Given a weighted, complete graph G, find a minimum weight tour.
  - TSP_e :
    - Given a weighted, complete graph G, find the weight of a minimum weight tour.
  - TSP_d :
    - Given a weighted, complete graph G, and a number k, find whether the weight of a minimum weight tour is less than k.

Optimization Problems – Formulations [3]

- Claim:
  - Given an optimization problem π
    - the construction version (π_c) is at least as hard as the evaluation Version (π_e)
    - which in turn is at least as hard as the decision version (π_d)
  - i.e.
    - $\pi_d \precsim \pi_e \precsim \pi_c$
- Observation:
  - If the decision version of a problem is $\mathbb{NP}$-hard, then so is the construction version!

Optimization Problems – Formulations [4]

- Given an optimization problem π **= <I, SOL, m, goal>** its decision Version (π_d) can be formulated in two ways :
  - Formulation 1:
    - Given an input instance **x**, and a threshold value **k** decide whether the optimal measure is bounded by **k**
      - i.e. Is **m*(x) <= k** ? (if **goal** is *min*)
  - Formulation 2:
    - Given an input instance **x**, and a threshold value **k**, decide whether there exists a feasible solution whose measure is bounded by **k**
      - i.e. Is there a **y** s.t. **y ϵ SOL(x)** and **m(x,y) <= k** ? (if **goal** is *min*)
- Exercise:
  - Prove that both these formulations are equivalent!

Optimization Problems – Formulations [5]

- Example: TSP_d
  - Formulation 1 :
    - Given a weighted, complete graph G, and a number k, find whether the weight of a minimum weight tour is less than k.
  - Formulation 2:
    - Given a weighted, complete graph G, a number k, find whether there is a tour of weight less than k.

$\mathbb{NP}$-hard Optimization problems - Examples

- The following optimization problems are $\mathbb{NP}$-hard :
  - Min Vertex Cover
  - TSP
  - 0,1 Knapsack
- because their decision versions are $\mathbb{NP}$-hard.