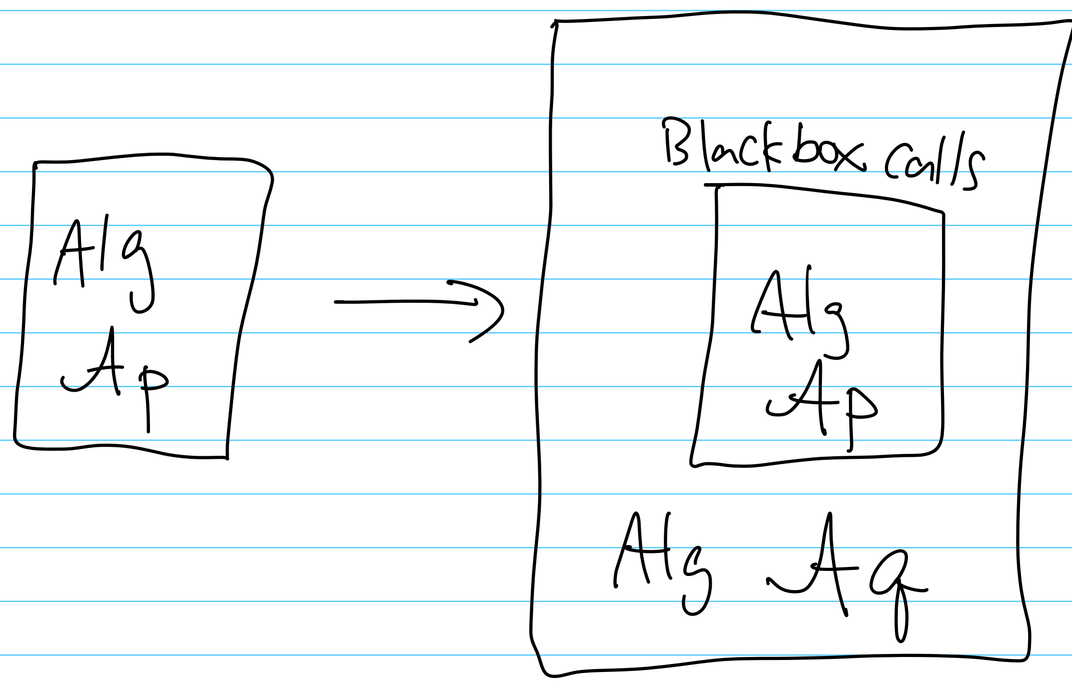
ECS 122A Lecture 14 Jasper Lee

Reductions

Suppose we have an alg A_p that solves Problem P,

can we use A_p to construct A_q for Problem Q?



A key algorithmic concept

Examples:

- Binary search application
- HW 3 Problem 3.2

Problem Q:

Find min k s.t. k is "feasible"

"Feasible": If k is feasible then $k' \geq k$ is also feasible.

Problem P:

On input k, determine if k is "feasible".

Reducing problem Q to P:

$\mathcal{A}_Q$:

Binary search over k in $[0 .. k_{max}]$

$L \leftarrow 0$; $R \leftarrow k_{max}$;

while $L \leq R$

$m \leftarrow \lfloor \frac{L+R}{2} \rfloor$

Call $\mathcal{A}_P$ to check if m is feasible.

If yes, $R \leftarrow m$

no, $L \leftarrow m+1$

} might need to fix loop termination condition

HW 3 Problem 3.2

Intended as DP, but can solve by Dijkstra's

Setup: Lineland road trip

motels

start					end
$0 = x_0$	loc: x_1	x_2	...	x_n	$x_{n+1} = 5000$
	cost p_1	p_2		p_n	$p_{n+1} = 0$

Can only drive 200 miles before resting.

Goal: Compute min cost to get from x_0 to x_{n+1}

Alg:

Construct directed weighted graph $G = (V, E)$

$V = \{x_0, x_1, \ldots, x_n, x_{n+1}\}$

$E = \{x_i \to x_j \mid i < j$ and $x_j - x_i \leq 200\}$ weight p_j

Run Dijkstra's to get min dist from x_0 to x_{n+1} on G

Correctness: min itin cost $= dist_G(x_0, x_{n+1})$

Proof: (2 directions usually)

$\geq$: Consider any min cost road trip itin

$0 = x_0 \to x_{i_1} \to x_{i_2} \to \cdots \to x_{n+1} = 5000$

Cost of itin $= p_{i_1} + p_{i_2} + \cdots + p_{i_k}$

where $k =$ # motels.

$\exists$ path from $x_0 \overset{G}{\rightsquigarrow} x_{n+1}$ w/ same cost.

$x_0 \xrightarrow{p_{i_1}} x_{i_1} \xrightarrow{p_{i_2}} x_{i_2} \xrightarrow{p_{i_3}} \cdots \xrightarrow{0} x_{n+1}$

This is a path on G b/c $x_{i_j} \xrightarrow{p_{i_{j+1}}} x_{i_{j+1}}$

is an edge on G, since

$x_{i_j} \to x_{i_{j+1}}$ on Lineland has distance

≤ 200.

So min itin cost = length of corresponding path on G

$\geq dist_G(x_0, x_{n+1})$

$\leq$: Consider any shortest path on G

$$x_0 \xrightarrow{p_{i_1}} x_{i_1} \xrightarrow{p_{i_2}} \cdots \xrightarrow{0} x_{n+1}$$

$\exists$ itin $x_0 \to x_{i_1} \to \cdots \to x_{n+1}$

(essentially same arg as above)

So $dist_G(x_0, x_{n+1})$ = cost of corresponding itin

$\geq$ min itin cost

$\square$

Computational Hardness

Informal Question:

Suppose we can use alg for Problem P to solve Problem Q

and we know/conjecture Problem Q is hard,

What can we say about Problem P?

Informal Answer: Problem P should also be hard.

(If we can solve P then we can solve Q,

$\Leftrightarrow$ if can't solve Q, then can't solve P)

Notion

Take ECS 120 for formal defn

NP-hard: If $P \neq NP$, then no polytime alg for a problem that is NP-hard.

How to show a Problem P is NP-hard?

☆ Find a problem Q known to be NP-hard, and find alg A_Q that

1. calls A_P blackbox poly many times
2. Does only polynomial extra work.

use HARDER problem alg to solve EASIER problem

(ECS 120 will talk about this slightly differently)

Known NP-hard problems

Graph:

Independent set

Given undirected unweighted G,
Find max # of vertices s.t.
no pair is connected by an edge.

Clique:

Given " G, find max # of
vertices s.t.
all connected to
each other

Many others

OG: (Cook-Levin Theorem)

SAT: Given Boolean Formula
with n variables $x_1, \dots, x_n$,
determine if $\exists$ asgn $x_i \mapsto T/F$
s.t. Formula evaluates to T.

Bonus: Assume NP-hardness of Ind Set, prove hardness of clique.

Alg for Ind Set:

1. Given $G=(V,E)$, construct
$$\overline{G} = (V, (V\times V)\setminus E)$$
(every edge becomes non-edge and vice versa)

2. Return MaxCliqueSize$(\overline{G})$.
(call to alg for Clique)

Correctness: Max Ind Set Size of G
= Max Clique Size of $\overline{G}$

$\leq$: Consider a max size ind set S of G.
for all $u, v \in S$, no edge (u,v) in G
so $(u,v) \in \overline{G}$

so S is clique in $\overline{G}$.

So Max ind set size of G
= size of some clique in $\overline{G}$

$\leq$ Max clique size of $\overline{G}$.

$\geq$: Consider a max size clique S of $\overline{G}$.

By symmetric argument as above,

S is ind set in G.

So Max clique size in $\overline{G}$

= size of some ind set in G

$\leq$ Max ind set size in G

//