04/02 350 Lec 1

1. Intro.

a. common techniques
- Divide & Conquer (分治)
- Greedy (贪心)
- dynamic programming (动态规划)

(Recursion)

(+ exposure to complexity theory.)

Alg. examples:
- Quick SORT (SORTING Alg's) Divide & Conquer
- Dijsktra Alg. (shortest PATH) Greedy
- KMP alg (string). (Bellman-Ford) ↓ Dynamic programming
- (BFS/DFS) → Prim (MST) Kruskal (Greedy)

what I'll do:

→ Modern Algorithms.
- Randomized algorithms.
- Approximation algorithms
- online/streaming algorithms.

1. Randomized alg's.

a. warm-up.

Computation needs resources

- time
- space. ✓

Randomness is a resource.

Computational

^_^ often extremely simple Alg's.

^_^ outperform deterministic alg's. on time/space

-_- fail sometimes.

prob. of failure $\to 0$ $\frac{1}{2^{100}}$

vs. prob of computer hit by a meteor $\sim \frac{1}{2^{60}}$

b. examples

- Primality testing (素数判定)

  Given: integer $N > 0$.

  $\text{len}(N) = \log N = n$

  (binary rep.)

  Goal: Is N prime?

  → direct alg's: test $2, 3, 4, \ldots, \sqrt{N}$?

  $O(N) = O(2^n)$

  → 70's: efficient randomized alg's

  [Miller-Rabin / Solovay-Strassen]. poly(n)

  → [AKS '04]: poly-time det. time

  ✡ higher poly, complicated!

  Real apps.

- Polynomial identity testing (PIT).

  → poly-time det. alg unknown

  ^^ ∃ efficient rand. alg.

  Simple

- Matrix-Product checking.

Given: 3 $n \times n$ matrices A, B, C.

Goal: decide if $AB = C$?

→ direct alg:
- $A \cdot B$ matrix mult
- compare w/ C (entry-wise). $O(n^2)$

M.M alg's: naïve $O(n^3)$

Strassen $O(n^{2.81})$
(Divide & Conquer)

$O(n^{2.37188})$ 2022

$\Omega(n^2)$ lower bound

$(O(\cdot), \Omega(\cdot))$

→ Rand. Alg: $O(n^2)$ alg.

A: on input A, B, C.

- Sample $r \in \{0,1\}^n$ unif. at rand.
- Output: $A(B \cdot \begin{pmatrix} r \\ 1 \end{pmatrix}) \stackrel{?}{=} C \cdot \begin{pmatrix} r \\ 1 \end{pmatrix}$

Time:

$B \begin{pmatrix} r \\ 1 \end{pmatrix} \to r' : O(n^2)$

$A \begin{pmatrix} r' \\ 1 \end{pmatrix} \to r'' : O(n^2)$

$C \begin{pmatrix} r \\ 1 \end{pmatrix} \to r''' : O(n^2)$

$r'' \stackrel{?}{=} r''' : O(n)$

$\Rightarrow$ Total time: $O(n^2)$

error example: $r = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$

- $AB = C$: always correct.
- $AB \neq C$: $\Pr_r [\underbrace{AB \cdot r = C \cdot r}_{\text{Bad event}}] \leq$ ____.

04/03 450 Lec 2

1. Probability Review

a. Basics

- **Sample space** Ω : set of all possible outcomes of a random experiment.
  - A single coin toss: $\Omega = \{H, T\}$ (H/1, T/0)
  - Two coin tosses: $\Omega = \{HH, TT, HT, TH\}$
  - Roll a 6-sided die: $\Omega = \{1, \dots, 6\}$.
- **Def.**: an **event** E is any **subset of** Ω

  E: getting **at least one** HEAD in **2 COIN TOSSES**.

  $\Omega = \{HH, TT, HT, TH\}$

  $E = \{HH, HT, TH\}$
- **DEF**: $E \& F$ **mutually exclusive** if $E \cap F = \emptyset$
  - E: roll an even number $\{2, 4, 6\}$
  - F: roll an odd number $\{1, 3, 5\}$
- consider a function:

  $P: \Omega \longrightarrow [0, 1]$

  $w \longmapsto P(w)$ $\quad \Pr(w)$

- Ex: $\Omega = \{H, T\}$

what do you want P to be?

- fair coin: $P(H) = P(T) = \frac{1}{2}$.
- Biased coin: $P(H) = 0.85$

$P(T) = 0.15$

Obs: $P(H) + P(T) = 1$.

- Ex: 2 coin tosses: $\Omega = \{HH, TT, HT, TH\}$.

→ $P(HH) = P(TT) = P(HT) = P(TH) = \frac{1}{4}$

→ $E = \{HH, TT\}$.

$$P(E) := P(HH) + P(TT) = \sum_{\omega \in E} P(\omega)$$

$$= \frac{1}{2}$$

- **Def:** A probability space is (Ω, P)
  - Ω: sample space
  - P: prob. function

$$\Omega \to [0,1]$$

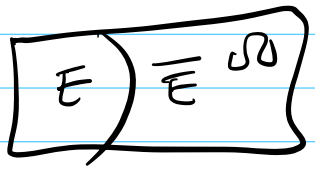
Axioms of Probability

- $P(\omega) \geq 0 \quad \forall \omega \in \Omega$
- $\sum_{\omega \in \Omega} P(\omega) = 1$
- if E & F mutually exclusive

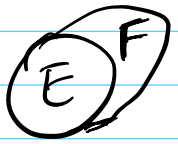
$$P(E \cup F) = P(E) + P(F)$$

· Cor:

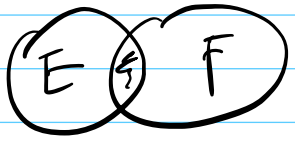
- $P(\bar{E}) = 1 - P(E)$ (Complement)



- if $E \subseteq F$: $P(E) \leq P(F)$



- $P(E \cup F) = P(E) + P(F) - P(E \cap F)$



- equally Likely outcomes

$$\forall w \in \Omega: P(w) = \frac{1}{|\Omega|}$$

$$\Rightarrow P(E) = \frac{|E|}{|\Omega|}$$

Ex. Toss a coin 100 times

what's prob. of 50 H's ?

A: $\frac{1}{2}$ B: $\frac{1}{2^{50}}$ C: $\binom{100}{50} / 2^{100}$ ✓

$|\Omega| = 2^{100}$

$|E| = \binom{100}{50}$

$\binom{100}{50} = C_{100}^{50}$

2. MPC

Given: A, B, C $n \times n$

Goal: ? $A \cdot B = C$

A:
choosen $r \leftarrow \{0,1\}^n$
check $A(B(r)) \stackrel{?}{=} C \cdot (r)$

If: $AB = C$: ~~A~~always output (YES) =

$AB \neq C$: incorrect $ABr = C \cdot r$
[e.g. $r = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$]

Claim: $AB \neq C$.

$$\Pr_{r \leftarrow \{0,1\}^n}(ABr = Cr) \leq \frac{1}{2}$$

(BAD event)

Pf: Let $D = AB - C$ $(AB \neq C)$

$\neq 0$

Promise: $\exists i,j,\ d_{ij} \neq 0$

$$D \cdot r = \begin{pmatrix} & & & \\ & & \vdots & \\ - & - & d_{ij} \neq 0 & - \\ & & \vdots & \end{pmatrix}_{n \times n} \begin{pmatrix} r \\ \vdots \\ r \end{pmatrix}_{n \times 1} = \begin{pmatrix} z \\ \vdots \\ z_i \\ \vdots \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

$$z_i = \sum_{k=1}^{n} d_{ik} \cdot r_k = 0$$

$$\Leftrightarrow \underbrace{d_{ij}}_{\neq 0} r_j = -\sum_{k \neq j} d_{ik} r_k$$

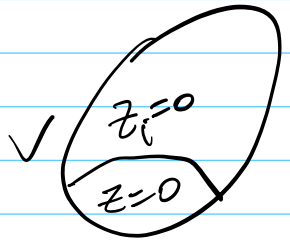
$$\Leftrightarrow r_j = \frac{-\sum_{k \neq j} d_{ik} r_k}{d_{ij}} = t$$

$$\star \Pr[r_j = t] = \frac{1}{2} = \Pr[z_i = 0]$$

(t independent of r_j)

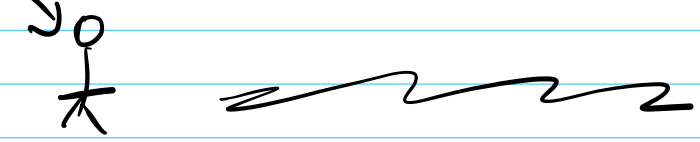
To fail: $\forall i \; z_i = 0$

$$P(z=0) \leq P(z_i = 0) = \frac{1}{2}$$



3. Finger printing.

$x \in \{0,1\}^n$ $\quad x \stackrel{?}{=} y \quad$ $y \in \{0,1\}^n$



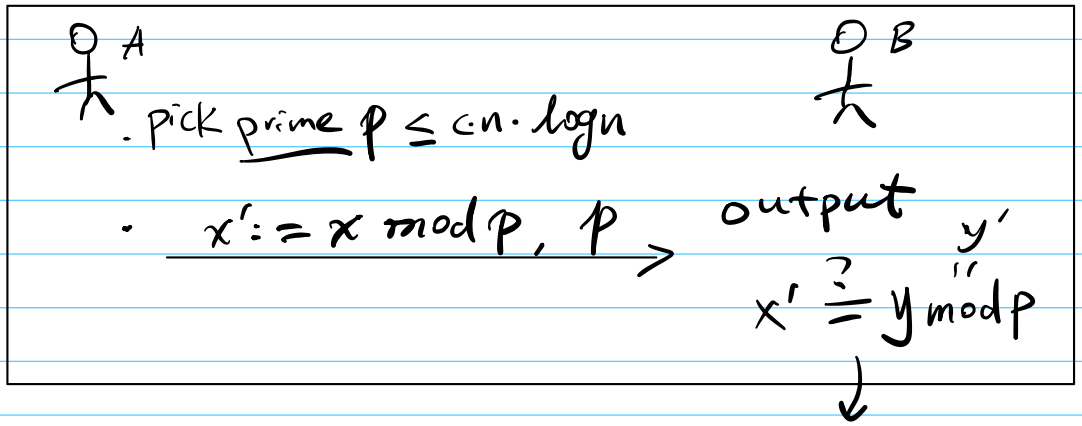
Alice $\qquad$ Bob

Given: x & y

Goal: decide if $x = y$ w/ few bits exchange

→ Det: space $\Theta(n)$.

→ Randomized: $O(\log n)$ bits



Analysis.

if $x = y$: $x' = y'$ always holds.

$x \neq y$: fail: $x \bmod p = y \bmod p$

$\Leftrightarrow p \mid (x-y)$

$\Pr_p(p \mid x-y) \le$ _____ .

04/09 350 Lec 3

FACT (Prime number theorem)

$|\{p : p \le a \text{ & prime}\}| \sim a/\log a$

$\Rightarrow$ # primes $\le c \cdot n \cdot \log n$ $(a = c \cdot n \cdot \log n)$

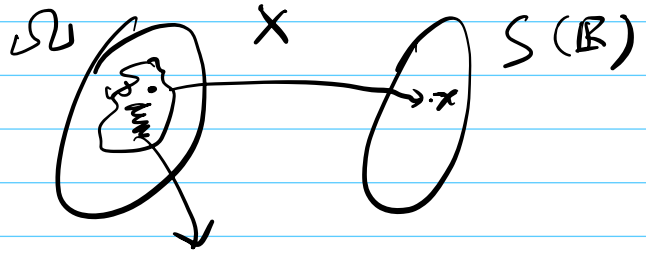
$\sim \frac{cn \log n}{\log(cn \log n)} \sim c \cdot n$

$\Rightarrow \Pr_p[p \mid x-y] \le \frac{n}{cn} \le \frac{1}{c} \le \frac{1}{4}$ if $c \ge 4$ #

1. Random variables

a. Basics

- DEF: $X : \Omega \to S(\mathbb{R})$



"$X = x$" : event $E := \{w : X(w) = x\}$.

- Expectation (期望) : weighted average.

$$\mathbb{E}[X] = \sum_{x \in S} Pr(X=x) \cdot x$$

★ linearity of expectation (LOE)

$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

- Ex: Bet game
  - You choose a card from a deck.
  - I pay you ¥ { 5 ♡ ; 0 O.W.
  - X : your earning.

$\Omega = \{♡, \overline{♡}\}$

$X : \Omega \to \mathbb{R}$

$\omega \mapsto X(\omega) \in \{0, 5\}$

$\mathbb{E}[X] = \sum Pr(X=x) \cdot x$

Ω	X	Pr(X=x)
♡	5	1/4
$\overline{♡}$	0	3/4

$$= \Pr(X=5)\cdot 5 + \Pr(X=0)\cdot 0$$

$$= \frac{1}{4}\cdot 5 = 1.25$$ ✱

Ex. Same setup.

- play it 100 rounds
- Y: total earning.
- $\mathbb{E}[Y]$:

Let X_i = earning in i^{th} round.

$i = 1, \cdots 100$

OBS: $Y = X_1 + \cdots + X_{100}$

$\forall i \quad \mathbb{E}[X_i] = 1.25$

$$\Rightarrow \mathbb{E}[Y] = \mathbb{E}[X_1 + \cdots + X_{100}]$$

(LOE) $$= \mathbb{E}[X_1] + \cdots + \mathbb{E}[X_{100}]$$

$$= 100 \times 1.25 = 125.$$ ✱

b. useful R.V.s

- Bernoulli R.V. $\longleftrightarrow$ biased coin toss

$$X = \begin{cases} 1 & p \\ 0 & (1-p) \end{cases}$$

$$\mathbb{E}[X] = p\cdot 1 + (1-p)\cdot 0 = p$$

- Binomial R.V. # of heads in n coin tosses w. bias p

$X \sim Bin(n, p)$

($\sim$: X has probability distribution)

$$\Pr[X=k] = \binom{n}{k} p^k \cdot (1-p)^{n-k}$$

$k = 0, 1, \cdots, n$

$$\mathbb{E}[X] := \sum_{k=0,\cdots,n} \Pr[X=k] \cdot k = \cdots$$

$$= n \cdot p$$

(Ex. use LoE)

- Ex: (Return phones).

$X :=$ # student get their own phone.

students $= n$.

$\mathbb{E}[X] = ?$

$i = 1, \cdots, n$

$$X_i := \begin{cases} 1 & \text{if student } i \text{ gets own phone} \\ 0 & \text{o.w.} \end{cases}$$

OBS:

$$X := \sum X_i$$

$$\mathbb{E}[X] = \mathbb{E}[\textstyle\sum X_i] \overset{\text{LOE}}{=} \sum_i \mathbb{E}[X_i]$$

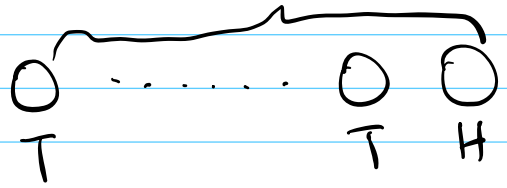
$$\forall i \quad \mathbb{E}[X_i] = \Pr[X=1] \cdot 1 + \Pr[X=0] \cdot 0$$

$$= \frac{1}{n}$$

$$\Rightarrow \mathbb{E}[X] = \sum_i \mathbb{E}[X_i] = n \cdot \frac{1}{n} = 1$$

#

- Geometric R.V. X



biased coin H w.p. p

X: # tosses till first see "H".

$X \sim Geo(p)$

Claim: $\Pr[X=n] = (1-p)^{n-1} \cdot p$

$\forall n = 1, 2, \ldots$

Claim: $\mathbb{E}[X] = \sum_{n=1}^{\infty} \Pr[X=n] \cdot n$

$$= 1/p$$

#

---

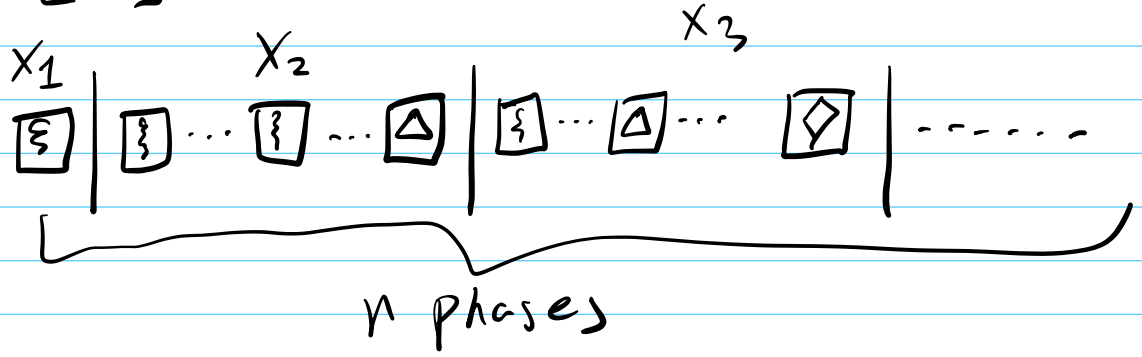
2. Coupon Collector Problem.

a. Description:

- n: coupons
- every time get one coupon uniformly at random
- $X :=$ # of purchases (open boxes) to collect at least one copy

of each coupon

? $\mathbb{E}[X]$.



$X_i :=$ already had $(i-1)$ coupons.

boxes to get a new coupon

$$X = \sum X_i = X_1 + \cdots + X_n$$

$X_i \sim \text{Geo}(p_i)$

(Geometric R.V.).

$i=2: p_2 = 1 - \frac{1}{n} = \frac{n-1}{n}$

$i=3: p_3 = 1 - \frac{2}{n} = \frac{n-2}{n}$.

$\vdots$

$i: p_i = 1 - \frac{i-1}{n} = \frac{n-(i-1)}{n}$

$\vdots$

$$\mathbb{E}[X_i] = 1/p_i = \frac{n}{n-(i-1)}$$

$$\Rightarrow \mathbb{E}[X] = \mathbb{E}[\sum X_i]$$

$$= \sum_{i=1}^{n} E[X_i]$$

$$= \sum_{i=1}^{n} \frac{n}{n-(i-1)}$$

$$= \frac{n}{n} + \frac{n}{n-1} + \cdots + \frac{n}{1}$$

$$= n \cdot \sum_{k=1}^{n} \frac{1}{k}$$ (调和级数)

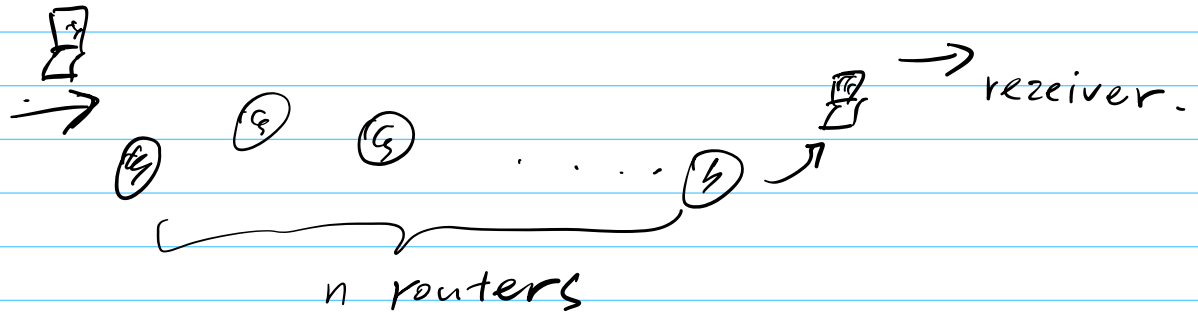
$H(n)$

Fact: $\ln n \le H(n) \le \ln n + 1$

$\Rightarrow E[X] \approx n \cdot \ln n$ ∎

04/10 350 Lec 4

0. Coupon collector application



Goal: Receiver wants to know all n routers

- Package: has space for one name & counter

Idea: sample a uniform router record its name.

⇒ Coupon collector problem
$(n \ln n)$ packages suffice in expectance

??? networking setting ☆

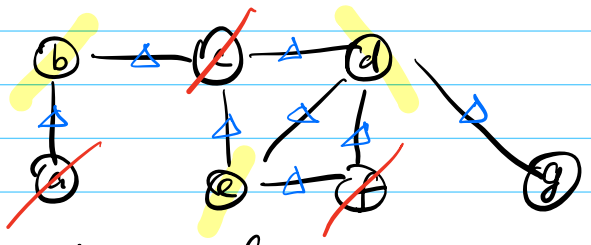
1. → Approximation algorithms.

: Intro: vector cover (node/vertex)

a. Basics: Graph $G = (V, E)$ (edges)

- DEF: A vector cover $S \subseteq V$ is a subset of V s.t. touch all edges.

Ex:



- V itself ✓
- S=(a, c, f, d) ✓
- (b, d, e) optimal VC

- Vertex cover

Given: $G = (V, E)$

Goal: find vertex cover S of minimum size.

what's known?

→ Brute force: check all subsets of V

$|V| = n$, $O(2^n)$

→ NP-hard: unlikely to admit a poly-time algorithm.

b. Approximation alg. for VC.

→ Greedy strategy:

Choose one that appears to be beneficial at the moment

1st attempt:

pick vertex that touches most edges.

App-VC1: on input $G=(V,E)$.

for $v \in V$ (in descending order of degrees (# edges in/out))

- add $v \in S$ (VC candidate)
- delete v & neighbors from G

- Analysis.
  - correctness: S will be VC ? ✓
  - optimality:

$$\exists G \text{ s.t. } |S| = \Omega(\log n \cdot OPT)$$

- 2nd attempt:

App-VC2: on input $G=(V,E)$

while some edge $\{u,v\}$ is uncovered

add both u, v to T (VC candidate)

output T

→ correctness ✓ (by termination condition)

→ optimality ?

claim: $|T| \leq 2 \cdot OPT$ (OPT: size of min VC)

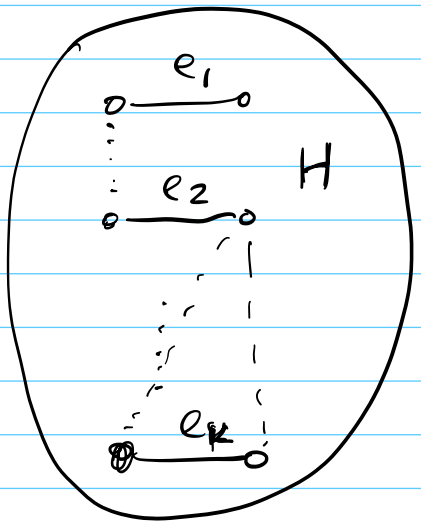
Pf:

Let $\{e_1, e_2, \dots, e_k\}$ be the edges chosen during the alg.

$|T| = 2k$.

Suffice to show $OPT \geq k$

G



H: is subgraph of G.

At least pick one node from each of k edges in optimal VC.

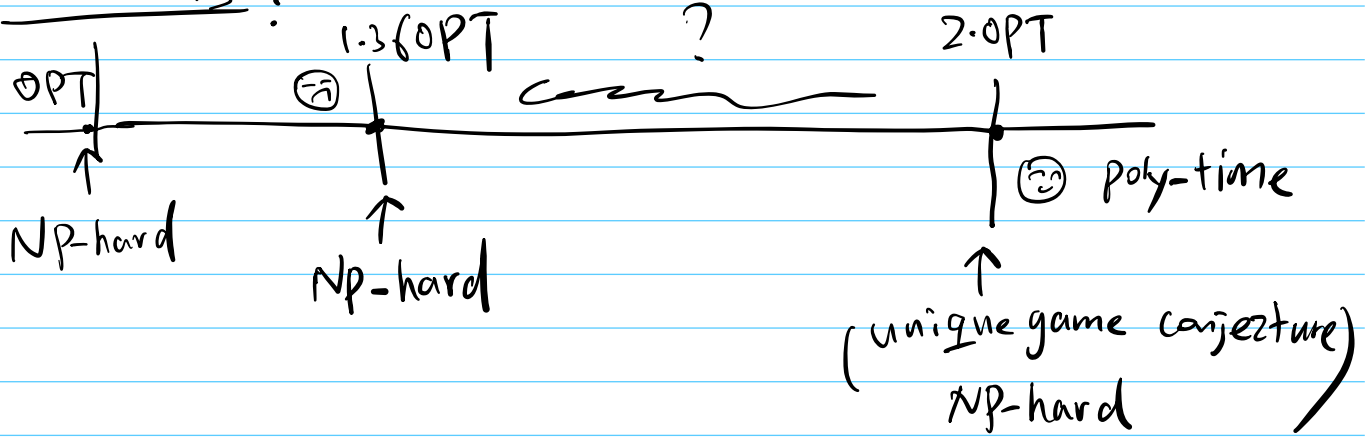
$OPT \geq k$

$\Rightarrow |T| \leq 2 \cdot OPT$.

(Approximation factor) #

$\Rightarrow$ Tight. (example on board).

Remarks:



A more principaled Approach:

(Integer) linear proprogramming (ILP)

2. Linear programing.

a. Basics.

Ex1. $\max\ x_1 + 5x_2$ $(x_1, x_2 \in \mathbb{R})$

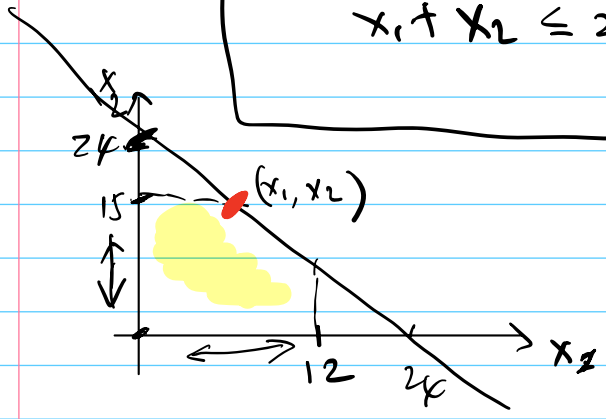
Subject to:

$0 \le x_1 \le 12$.

$0 \le x_2 \le 15$.

$x_1 + x_2 \le 24$.

→ linear constraints.



OBJ: $x_1 + 5x_2$

$= (x_1 + x_2) + 4x_2$

$\le 24 + 4 \times 15 = 84$

$x_1 = 9,\ x_2 = 15$

Ex2: max $x_1 - x_2$ $(x_1, x_2 \in \mathbb{R})$

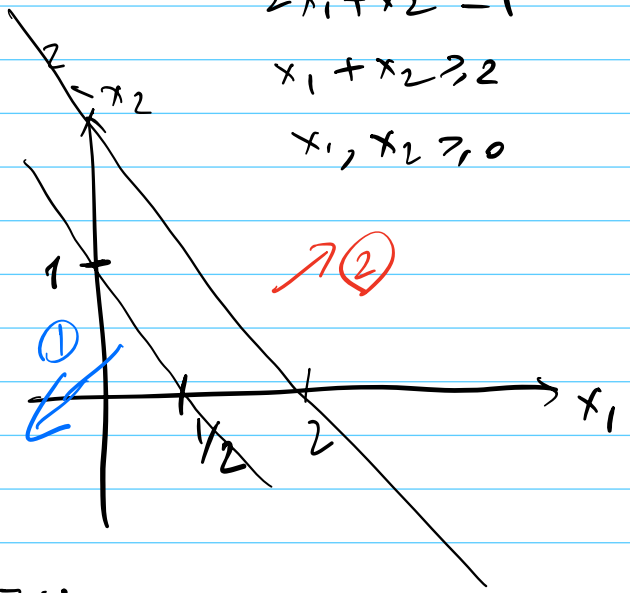
subject to:

$2x_1 + x_2 \leq 1$

$x_1 + x_2 \geq 2$

$x_1, x_2 \geq 0$

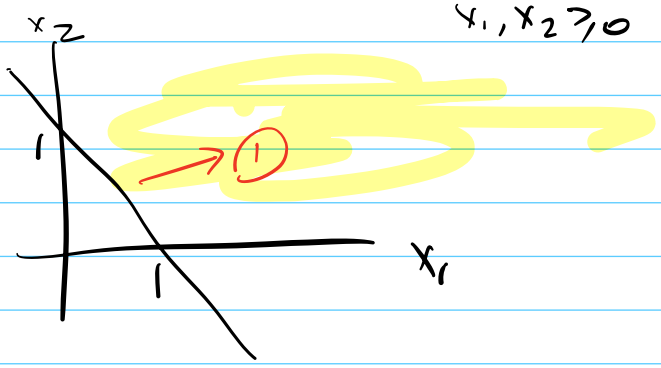
(infeasible)



Ex3. max $2x_1 + x_2$ (unbounded)

Subject to: $x_1 + x_2 \geq 1$

$x_1, x_2 \geq 0$



b. LP algorithms. (for feasible instances).

- Simplex alg. (George Dantzig 1947)

→ Running time: ☹ worst-case exp. time.

☺ super fast real world

- Poly-time LP Alg's.
  - → Ellipsoid alg. [Khachiyan '1979].
    (NOT competitive in praztice)
  - → Interior point alg. [Karmakar '1984]
    Naredra

N.B. Commecial solvers
solve LP w/ millions of variable
& constraints ✓