LECTURE #2

PROBLEMS:

State-space: Explore the state-space based on input data
Reach a final state, which allows one to make a decision: $\in \{0,1\}$
How one explores the state space is based on a set of computational rules?
↳ "Stuff doing stuff to other stuff."

~~Q/=ΣB~~

Resources: How much stuff do you need?
{ Time, Space, Data, ... Bandwidth, processors, ... → Complexity.

Q = Finite Set of states
Σ = Finite Set of symbols
$s_0 \in Q$ = Initial state
$b \in \Sigma$ = Blank symbol
$A \subseteq Q$ = Set of Final states
$\delta \subseteq (Q \setminus A \times \Sigma) \times (Q \times \Sigma \times \{L, R\})$ = Transition Relation

If the transition relation is one-to-one (bijective) (graph of a transition function) the computation is

DETERMINISTIC; otherwise, NONDETERMINISTIC.

↓

Computation Tree Model. (Branching Time Logic)
↳ Most Creative Guesser
ALWAYS GUESS THE CORRECT "PATH"

NONDETERMINISM

Guess-&-Check
"Abeland-Eloise Game"

"For every instance of the problem, there exists a guessed soln, so that it can be checked that the guessed soln satisfies the desired decision."

$$\forall x \quad \exists y \quad P(x,y)$$

Input — Guess — Deterministic Check.

$\forall$belard - $\exists$loise - Game.

Quantifier-Elimination: How powerful is the ability to guess?

Determinism	vs	Nondeterminism.
FSA	vs	NFSA
P	vs	NP → Core Question in CS.
DTM	vs	NDTM.

DTIME$(f(n))$ ← A problem of inputsize n requiring $O(f(n))$ computation time to solve using DTM.

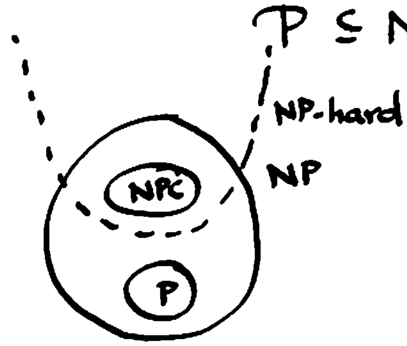
$$P = \bigcup_{k \in \mathbb{N}} DTIME(n^k)$$

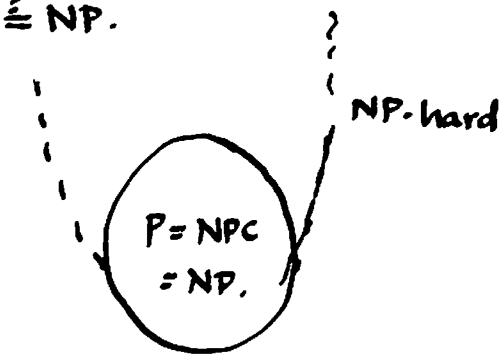
NTIME$(f(n))$ ← A problem of inputsize n requiring $O(f(n))$ computation time to solve using NDTM.

$$NP = \bigcup_{k \in \mathbb{N}} NTIME(n^k)$$

$$p(x,y) \in P \Leftrightarrow \exists y, |y| \le |x|^c \; p(x,y) = q(x) \in NP$$

$P \subseteq NP \quad , \quad P \stackrel{?}{=} NP.$

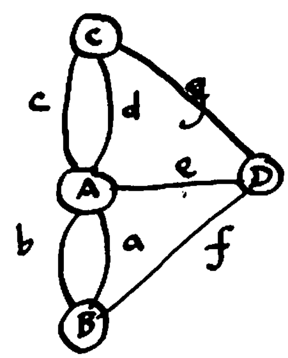




Historical Digression:

18^{th} Century königsberg (Kaliningrad).

Seven Bridges straddling river Pagel



$\boxed{\text{Can you walk through Königsberg in a way that crosses each bridge over Pagel exactly once?}}$

Generalize:

DATA: $G = (V, E)$ connected undirected graph with $E \subseteq V \times V$.

DESIDERATA: Does G have a walk (trail, non-simple path)

$$v_1, e_{i_1}, v_2, e_{i_2}, \ldots, v_n, e_{i_n}, v_{n+1} \in EC$$

$$e_{i_j} = (v_i, v_{i+1}) \in E$$

$e_{i_j} \neq e_{i_k}$ (all edges are distinct $\Rightarrow$ trail)

Closed Trail = Tour $\Longrightarrow$ $v_1 = v_n$ (vertices need not be distinct $\Rightarrow$ non-simple path)

that includes every edge

$$\forall_{e \in E} \quad e \in EC ?$$

CLOSED TRAIL THAT INCLUDES EVERY EDGE
≡ EULERIAN CYCLE

A graph containing an Eulerian Cycle = Eulerian Graph.

Leonhard Euler: (1736).

Theorem: A connected graph contains an Eulerian Cycle iff every vertex has even degree.

(Proved by Carl Hierholzer 100 years later).

Corollary: If exactly two vertices have odd degree, it contains an Eulerian path but not an Eulerian cycle.

$$G = \text{Eulerian} \quad \text{iff} \quad \forall_{v \in V} \ \deg(v) \equiv 0 \mod 2$$

$$\text{iff} \quad \forall_{v \in V} \ |Adj(v)| \equiv 0 \mod 2$$

$$\text{iff} \quad \forall_{v \in V} \ \sum_{\substack{u \neq v \\ u \in V}} A_{uv} \equiv 0 \mod 2$$

A: Adjacency Matrix

$$\text{Eulerian} \begin{Bmatrix} \text{Path} \\ \text{Cycle} \end{Bmatrix} \in \mathcal{P}$$

Proof: One direction is trivial: If you enter a vertex by an edge incident on it, then you must exit by another edge incident on it.

Other direction → Proof by induction on the size of the graph.

→ FLEURY'S ALGORITHM.

FLEURY'S ALGORITHM

Let G' be the graph formed by the edges you have not yet crossed.
G' must have an Eulerian path (two vertices of odd degree: s and t)
Extend the trail $(G \setminus G')$ by selecting an edge (incident on s or t) so that removal of that edge from G' results in a graph G'' that is also Eulerian.

$$T(n) = T(n-1) + \text{poly}(n) \Rightarrow T(n) = O(n\ \text{poly}(n)) \in P.$$

Sir William Rowan Hamilton (1859)

Icosian Game: Walk around the edges of dodecahedron while visiting each vertex once and exactly once.

Generalize:
DATA: $G = (V, E)$ connected directed graph with $E \subseteq V \times V$.

DESIDERATA: Does G have a simple path

$$v_{\pi(1)}, v_{\pi(2)}, \cdots, v_{\pi(n)} \qquad \pi \in S_n, |V| = n$$

such that

$$\forall_i \ (v_{\pi(i)}, v_{\pi(i+1)}) \in E$$
$$(v_{\pi(n)}, v_{\pi(1)}) \in E \quad ?$$

A SIMPLE CYCLE CONTAINING EVERY VERTEX $\equiv$ HAMILTONIAN CYCLE.

A graph containing a Hamiltonian Cycle $\equiv$ Hamiltonian Graph

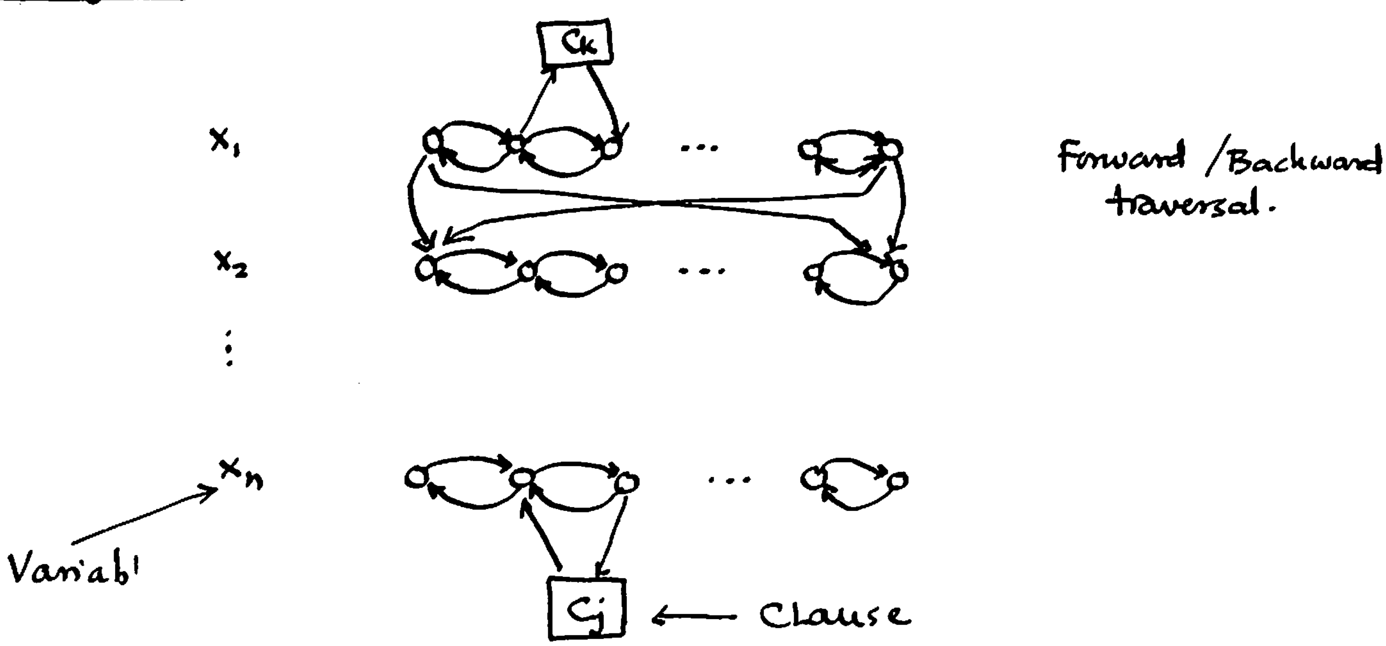
Graph, $G = (V, E)$ is Hamiltonian iff 13

$\exists_{\pi \in S_n} \; v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)}$ = Simple Cycle.

Hamiltonian $\left\{ \begin{matrix} \text{Path} \\ \text{Cycle} \end{matrix} \right\} \in NP$.

Can we find some way to characterize it, so that it is in P?

<u>Gadgets:</u>



3-SAT $\leq_p$ Hamiltonian Cycle.

Traveling Salesman Problem: (TSP)

<u>Data</u>: $G = (V, E)$ = Undirected weighted complete graph

$$E = V \times V$$
$$\omega : E \to \mathbb{R}^+$$

<u>Desiderata</u>: A Hamiltonian Cycle $C^* = v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)}$

$$C^* = \min_{\pi \in S_n} \sum_{i=1}^{n} \omega(v_{\pi(i)}, \pi(i+1)) \qquad \boxed{\pi(n+1) \equiv \pi(1)}$$

Metric TSP:

* ~~$w(e) \geq 0$~~

$$w(u,v) \geq 0; \quad w(u,v) = w(v,u);$$
$$w(t,u) + w(u,v) \geq w(t,v)$$

Positive, Symmetric, satisfying triangle inequality.

Euclidean TSP:

$$V \subseteq \mathbb{R}^d$$
$$v = (x_1, x_2, \cdots, x_d) = \text{d-dimensional point}$$

$$w(u,v) = \left(\sum_{i=1}^{d} (x_i - y_i)^2 \right)^{1/2}, \quad \text{where}$$
$$u = (x_1, x_2, \cdots, x_d)$$
$$v = (y_1, y_2, \cdots, y_d)$$

$$w(u,v) \geq 0, \quad w(u,v) = w(v,u) \quad \&$$
$$w(t,v) \leq w(t,u) + w(u,v).$$

Solution to Metric TSP:

a) Brute-force: Try all $n! = |S_n|$ possible permutations... Enumerate and Test.

$$n! = O(2^{n \ln n}) = T(n)$$

b) Dynamic Programming:

$C(S,j)$ = Shortest path starting at 1 visits all nodes in S and ends at j

$$C(S,k) = \begin{cases} d_{1,k}, & \text{if } S = \{1,k\}; \\ \min\limits_{\substack{m \neq k \\ m \in S}} \left[C(S \setminus \{k\}, m) + d_{m,k} \right], & \text{otherwise.} \end{cases}$$

$$T(n) = (n-1) \sum_{k=1}^{n-3} \binom{n-2}{k} + 2(n-1) \sim O(n^2 2^n)$$

c) Integer Programming

$V = \{0, \dots, n\}$

$x_{ij} = \begin{cases} 1, & \text{the path goes from node } i \text{ to node } j; \\ 0, & \text{otherwise.} \end{cases}$

$u_i =$ Auxiliary Variable $\in \mathbb{N}$
Integer valued - Intuition: $u_i = t$ the city visited in step t $[\pi(t) = i]$

$$\boxed{u_i - u_j + n\, x_{ij}} \xrightarrow{x_{ij}=0} = u_i - u_j \leq n-1 \quad \because 1 \leq u_i, u_j \leq n$$

$$\boxed{u_i - u_j + n\, x_{ij}} \xrightarrow{x_{ij}=1} = t - (t+1) + n = n-1$$

TSP:

$$\min \sum_{i=0}^{n} \sum_{\substack{j \neq i \\ j=0}}^{n} w_{ij}\, x_{ij}$$

$$0 \leq x_{ij} \leq 1 \qquad u_i \in \mathbb{Z} \qquad \forall i, j = 0, \dots, n$$

$$\sum_{i=0,\, i \neq j}^{n} x_{ij} = 1 \qquad \forall_{j = 0, \dots, n}$$

$$\sum_{j=0,\, j \neq i}^{n} x_{ij} = 1 \qquad \forall_{i = 0, \dots, n}$$

$$u_i - u_j + n x_{ij} \leq n-1 \qquad \forall_{\substack{i,j = 1, \dots, n \\ i \neq j}}$$

Bounds:

a) Lower Bounds:

Let C be a Hamiltonian Cycle in G with a total cost of c^*, which is minimal.

Let $e \in C$ be an edge in the Hamiltonian Cycle. Then $C \setminus \{e\}$ is a spanning tree of G.

$\Rightarrow$ If M is a MST (minimal spanning tree) of G.

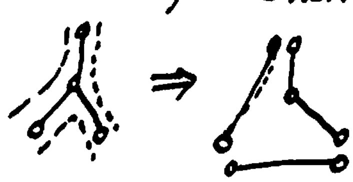
$$w(M) \le w[C \setminus \{e\}] \le c^*$$

b) Upper Bound (Metric TSP)

$$w(M) \le c^* \le 2\,w(M)$$

a) Traverse the tree twice to create a closed walk.

b) Short-~~circuit~~ cut around the vertices ~~to turn the~~ that are visited more than once to turn the walk to a path and shorten the distance (using triangle inequality).



c) $c^* \le 2w(M)$

c) Upper Bound (Metric TSP).

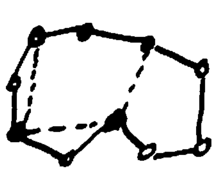
a) M = MST of G.

b) O: Odd degree vertices in M $\quad |O|$ = Even.

$\left[\because \sum_{v \in V} \deg_M(v) = 2(n-1)\right]$

c) $\mathrm{TSP}(O) \le c^*$

Complete Subgraph induced by O



C

Delete all vertices from C except the ones in O, and connect the subpaths by shorter path

d) M' = Minimum weight Perfect Matching in the complete subgraph induced by O.

e) TSP(0) can be broken up into two possible perfect matchings M_1' and M_2'

$$w(M') \leq \min\left[w(M_1'),\ w(M_2') \right] \leq \tfrac{1}{2}\left[w(M_1') + w(M_2') \right]$$
$$= \tfrac{1}{2}\, TSP(0) = \tfrac{1}{2}\, c^*$$

f) Combine M and M′ to create a multigraph H; H has even degrees and connected

$\Rightarrow$ H = Eulerian

g) Turn the Eulerian Cycle into a Hamiltonian Cycle by short-cutting [skip visited nodes] $\Rightarrow \hat{C}$

$$c^* \leq w(\hat{C}) \leq w(H) \leq w(M) + w(M')$$
$$\leq c^* + \tfrac{1}{2} c^* = \tfrac{3}{2} c^*$$

$$w(\hat{c}) \geq c^*$$

$$\boxed{\tfrac{2}{3}\, w(\hat{c}) \leq c^* \leq w(\hat{c})}$$