NP Complete problems

Traveling Salesman Problem (TSP)
- find a path that visits each city
- minimize the total length.
- "visit every vertex once"

\[ n = |V| \]

Search space is huge

\[ O(n!) \]

Naive

Dynamic programming SD

\[ O(2^n \cdot n^2) \]

Exponential

MST: Search space in the set of all possible spanning trees

G

Large search space is not the only criteria that make a problem hard.

Optimization problems
Optimization problems

$T \leftarrow \text{Input}$

$A(T) \rightarrow \text{optimal value} \rightarrow \min \rightarrow \text{solution}$

$\text{opt-Tsp : smallest tour of the vertices}$

Optimization version

Decision problem / Search problem

$T \leftarrow \text{Input instance}$

$u \leftarrow \text{Value}$

$A(T, u) \rightarrow \{0, 1\} \rightarrow \{\text{False, True}\}$

If there exists a solution $S$ such that
the optimal value for $S$ is $\leq u$ then return 1
else return 0

$\text{d-Tsp} (G, u)$:

decision Tsp does there exist a tour of all vertices, such that $\text{cost} (\text{tour}) \leq u$
Given an opt problem, we can convert it into a decision problem.

Any decision problem can be converted into an opt problem. How to generate potential "test" thresholds:

\[ v = 10 \quad \Rightarrow \quad v \rightarrow 1000 \]
\[ v = 50 \]

"Binary" search on unknown \( b \):

\[ v \]
\[ 1 = 2^0 \]
\[ 2 \]
\[ 4 \]
\[ 8 \]
\[ 16 \]
\[ 32 \]
1) find via "doubling" the interval that contains $b$
   \[ \log b \text{ steps or } k \text{ steps} \]
   \[ b = 760 \]

2) just do a regular "binary" search on the interval
   \[ [2^{k-1}, 2^k] \]
   \[ \text{binary search takes } k-1 \text{ steps} = O(\log b) \]

run $d$-TSP $O(\log b)$ time to find the optimal value $b$.

$\log b$ actually polynomial in the input size

opt $\equiv$ decision with a factor of $\log b$, $b$ is the optimal value.
Optimization problem \( \text{Opt} - TSP \)

how to verify if a solution is correct

a sol to \( TSP \): \( \{2, 1, 5, 3, 4\} \)

\[ \text{Cir} : 10 + 6 + 10 + 2 + 5 = 33 \text{ vs. optimal value?} \]

decision problem:

Solution \( \{2, 1, 5, 3, 4\} \)

threshold \( v = 20 \)

\[ \text{Cir} = 33 \geq 20 \]

return False / 0

\( \text{NP} \): Class of problems for which there exist a polynomial-time verification algorithm

\[ V(s, v) = \begin{cases} 1 & \text{if cir}(D) \leq v \\ 0 & \text{otherwise} \end{cases} \]

\( \text{P} \): Class of problems solvable in poly-time

\( \text{NP} \): "Verifiable"

\( \text{P} \subseteq \text{NP} \)
There exists a problem that is "not computable".

*Example:*

**Halting problem**

**HP (Algorithm A):**

1. If \( \text{HP}(A) \) halts then go to 1.

**Paradox (A):**

\( \text{HP(Paradox (A))} \)

**Proof system:**

Cannot be both consistent and complete.
\[ \Omega(?) = \frac{1}{2} \ln(2) = O(2^n, n^2) \]

What is the best possible lower bound?

\[ P \neq NP \]

**NP-Complete problems**

**NPC** : harder problems in NP.

Given a problem \( L \), we say \( \exists \) a reduction from \( L \rightarrow L' \), denoted \( L \rightarrow L' \);

\( y \rightarrow \) a polynomial function \( f \)

such that

- \( y \) is a solution for \( L \) if and only if
- \( f(y) \) is a solution for \( L' \)

Decision problem

\[ S \rightarrow f(s) \rightarrow L' \rightarrow \{ \text{1 if } f(s) \text{ is sol to } L' \}, \text{0 if } f(s) \text{ is not} \]
Suppose $L'$ takes polynomial time.

$$\Rightarrow L \in \text{P}$$

If $L' \in \text{P}$ and $L \rightarrow L'$, then $L \in \text{P}$.

**NP-complete (NPC):**

$L$ is a problem that is NP-complete if $L \in \text{NPC}$.
If even we find that $L \in \text{NPC}$ also belong to $P$

\[ \text{then } P = \text{NP} \]

$\text{P} \neq \text{NP} \quad \text{"accepted belief"}$

$\text{NPC} \leftarrow \text{TSP vs MST in P}$

$\text{constraint}$

\begin{itemize}
  \item \text{Eulerian Path vs Hamiltonian Path (Rudrata Path)}
\end{itemize}
Euler:
G: find a path that contains each edge only once
   visit all vertices once
Hamilton/Rudolph:
G: find a path that visits each vertex only once
   \[ \text{NP} \]

Find an Euler path:
Each edge should be used only once

There's no Euler path for this graph

Euler path exists if all vertices have even degree except possibly the source & sink vertex
G has to be connected
All intermediate vertices on an Euler path must have even degrees.

(1) Time Euler path solution
(2) Cycle decomposition of the S-T path
(3) Visit all cycles between S and T, their union will give you Euler path!