Lecture - 9 Christofede's Algorithm for TSP 2/3/2015

→ Christofede's Algorithm for TSP

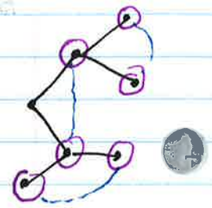
→ Dynamming Programming

1. Fibonacci Numbers
2. Max Window
3. Exact TSP is less than $O(n!) = O(n^n)$

Christofede's Algorithm

Black Boxes →

1. Compute MST ($\leq OPT$)
2. Let B be the odd degree nodes.
(MM) Compute Min weight perfect matching
Add to T (≤ 1.5 OPT)
3. Compute Eulerian Tour and Shortcut.



Original graph is complete

Claim: In any graph, the number of odd degree nodes is even.

Proof: By hand shake lemma,

Claim: (M M) has weight $\leq \frac{1}{2}$ OPT

"Matching:" A bunch of monogomous ~~edge~~

OPT: the answer to the travelling salesman problem.

→ If two matchings M M is min weight matching.
⇒ $MM \leq \frac{1}{2} OPT$

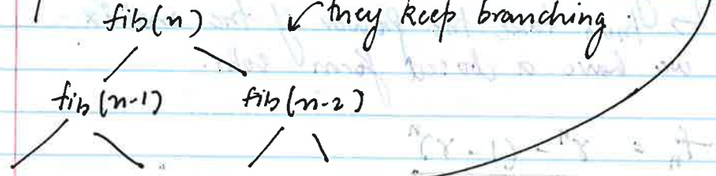
→ DYNAMIC PROGRAMMING: (doing recursion)

- Fibonacci Numbers.

$f_0 = f_0 = 1$
$f_n = f_{n-1} + f_{n-2}$

```
fib(n)
    if (n == 0 or 1)   return 1
    return fib(n-1) + fib(n-2)
```

Why exponential time?



Computed(n): Array[Bool] (false)(n)
Saved: Integers

```
fib(n)
    if (n == 0 or 1) return 1
    if (computed(n)) return saved(n)
    else ~~computed(n) = true~~
        saved(n) = fib(n-1) + fib(n-2)
        computed(n) = true
        return saved(n)
```

→ What you are really doing recursion & memoization.

Dynamic programming require you to fill in the array yourself.

```
fib(n) (Int)
f(0) = 1
f(1) = 1
for i = 2 : n
    fib(i) = fib(i-1) + fib(i-2)
```

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_n \\ f_{n-1} \end{bmatrix} = \begin{bmatrix} f_{n+1} \\ f_n \end{bmatrix}$$

↳ diagonalize the matrix.
↳ then take the power of the matrix.

so, we have a closed form soln.

$$f_n = \frac{\gamma^n - (1-\gamma)^n}{\sqrt{5}}$$

γ = golden ratio.

MAX WINDOW

→ given an array $\{a_i\}$ of real numbers, find i, j such that $\sum_{k=i}^{j} a_k$ is maximized.

Pre process: $S_i = \sum_{1 \to i} a_i$ → $O(n)$ time.

→ for all (i, j) → $O(n^2)$ time.
compute $\sum_{i \to j} a_k$ → $O(1)$ time

★ MAX WIN (i) = { maximum window ending at position i ★

MAX WIN $(\phi) = a_0$ ← trivial.

MAX WIN $(i) = \max(a_i, a_i + \text{maxwin}(i-1))$

$O(n) \to$ maxwindow $= \max_i$ maxwin(i)

EXACT SOLN. FOR TSP

let $T(S, j)$ = min weight path starting at 1 and visiting all nodes in S and ending at j.

$S \subseteq V$, $1 \in S$, $j \in S$, $1 \le j \le n$

claim I can rewrite this recursively.

Amount work: $\theta(n^2 2^n)$
Space: $\theta(n 2^n)$

final TSP answer: $\min_{i \neq 1} T(V, i) + d_{i1}$.

$$= \begin{cases} \text{if } |S| = 2, \quad S = \{1, j\} \qquad j = 2, \ldots, n \\ \quad \text{cost} \to = d_{1j}. \\ \text{else} \\ \quad \min_{\cancel{i \neq j}\ i \neq j} d_{ij} + T(S - \{i\}, i) \end{cases}$$

$$= \begin{cases} \text{if } |S| = 1, \quad S = \{1\} \qquad j = 2, \ldots, n \\ \quad = d_{1j} \\ \text{else} \\ \quad \min_{i \neq j} d_{ij} + T(S - \{i\}, i) \end{cases} \to O(n) \text{ work.}$$