Lecture 9: Christofides' Algorithm for TSP 2/3/2015

- Christofides' Algorithm for TSP
- Dynamic Programming

1. Fibonacci Numbers
2. Max Window
3. Exact TSP is less than \( O(n^4) \)

Christofides' Algorithm

1. Compute MST (\( \leq \text{OPT} \))
2. Let \( M \) be a matching of odd degree nodes.
   \( 2M \) is a maximum matching of odd degree nodes.
   \( M \) is a maximum weight perfect matching.
3. Compute Eulerian Tour and Shortcut

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

Proof: By hand shake lemma.

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

"Matching" A bonus of monomorphic edge

\[ \text{OPT} \]

\[ \text{MM} \text{ is } \text{min weight matching} \]

\[ \text{MM} \leq \frac{1}{2} \text{OPT} \]

Dynamic Programming: (doing recursion)

- Fibonacci Numbers
- \( f_0 = f_1 = 1 \)
- \( f_n = f_{n-1} + f_{n-2} \)

\[ \text{fib}(n) \]

- \( \text{fib}(n) \) if \( n = 0 \) or \( 1 \) return 1 return \( \text{fib}(n-1) + \text{fib}(n-2) \)

Why exponential time?

\[ \text{fib}(n) \]

- They keep branching

\[ \text{fib}(n) \]

- Computed(n) = Array bool[0..n](false)[n] saved = 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)
Dynamic programming requires you to fill in the array yourself.

\[
\text{fib}(n) \ (\text{Int}) \\
\begin{align*}
\text{fib}(0) &= 1 \\
\text{fib}(1) &= 1 \\
\text{for } i = 2 \text{ to } n: \\
\text{fib}(i) &= \text{fib}(i-1) + \text{fib}(i-2)
\end{align*}
\]

\[
\begin{bmatrix}
1 & 1 \\
0 & 1
\end{bmatrix}^n = \begin{bmatrix}
F_{n+1} \\
F_n
\end{bmatrix}
\]

[Diagonalize the matrix. Compute powermat in \(O(n^3)\) time.]

We have a closed form equation:

\[
f_n = \frac{Y^n - (1-Y)^n}{\sqrt{5}}
\]

\(Y = \text{golden ratio}\).

\[
\text{Exact Soln. for TSP:}
\]

Let \(T(s, j) = \min \text{ weight path starting at } s \text{ and visiting all nodes } i < j \text{ and ending at } j\).

Claim 1: We can compute this recursively.

Amount work: \(O(n^2 \cdot 2^n)\)

Space: \(O(n^2)\)

Final TSP answer: \(\min_{i \neq j} T(s, i) + \text{dist}_{i, j} \)