CS 294-8 Computational Biology for Computer Scientists **Spring 2003**

Lecture 1: January 21

Lecturer: Gene Myers *Scribe: Hao Chen*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

1.1 All Pairs Shortest Paths

The naive and obvious solution to All Pairs Shortest Path(APSP) problem is to run a Single Source Shortest Path algorithm from each starting vertex v. If the graph has arbitrary edge weights, it takes the Bellman-Ford algorithm $O(|E||V|^2)$ time to solve APSP. But there are better approaches.

1.1.1 Floyd-Warshall Algorithm: Dynamic Programming

Label the vertices $1, 2, \ldots, n$. Define $d^{(k)}(i, j)$ to be the length of a shortest path from i to j, using intermediate vertices from $\{1, 2, \ldots, k\}$ only. Obviously, $d^{(n)}(i, j)$ is the full problem.

. . .

1.2 Transitive Closure

Our goal is to achieve running time $O(M(n) \log n)$ for APSP where $M(n)$ is the time for $n \times n$ matrix multiplication. Let's see if we can achieve this for a simpler but related problem, namely *Transitive Closure*:

. . .

References

[AGM97] N. Alon, Z. Galil and O. Margalit, On the Exponent of the All Pairs Shortest Path Problem, *Journal of Computer and System Sciences* **54** (1997), pp. 255–262.

[F76] M. L. Fredman, New Bounds on the Complexity of the Shortest Path Problem, *SIAM Journal on Computing* **5** (1976), pp. 83-89.