Part VI

Interaction between algorithms and data structures: Case studies in geometric computation

Organizing and processing Euclidean space

In Part III we presented a varied sample of algorithms that use simple, mostly static, data structures. Part V was dedicated to dynamic data structures, and we presented the corresponding access and update algorithms. In this final part we illustrate the use of these dynamic data structures by presenting algorithms whose efficiency depends crucially on them, in particular on priority queues and dictionaries. We choose these algorithms from computational geometry, a recently developed discipline of great practical importance with applications in computer graphics, computer-aided design, and geographic databases.

If data structures are tools for organizing sets of data and their relationships, geometric data processing poses one of the most challenging tests. The ability to organize data embedded in the Euclidean space in such a way as to reflect the rich relationships due to location (e.g., touching or intersecting, contained in, distance) is of utmost importance for the efficiency of algorithms for processing spatial data. Data structures developed for traditional commercial data processing were often based on the concept of one primary key and several subordinate secondary keys. This asymmetry fails to support the equal role played by the Cartesian coordinate axes x, y, z, … of Euclidean space. If one spatial axis, say x, is identified as the *primary key*, there is a danger that queries involving the other axes, say y and z, become inordinately cumbersome to process, and therefore slow. For the sake of simplicity we concentrate on two-dimensional geometric problems, and in particular on the highly successful class of plane-sweep algorithms. Sweep algorithms do a remarkably good job at processing two-dimensional space efficiently using two distinct one-dimensional data structures, one for organizing the x-axis, the other for the y-axis.

Literature on computational geometry. Computational geometry emerged in the 1970s as the most recent major field of algorithm design and analysis. The extremely rapid progress of research in this field is reflected by the fact that the few textbooks do not yet present a comprehensive sample of the field but rather, describe distinct aspects. The following books, taken together, survey the field: [Ede 87], [O'R 87], [PS 85], and [Hof 89]

References

[Ede 87] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, EATCS Monographs on Theoretical Computer Science, Vol. 10, Springer-Verlag, New York, 1987.

[HNS 88] K. Hinrichs, J. Nievergelt, and P. Schorn, Plane-sweep solves the closest pair problem elegantly, *Information Processing Letters 26*(5), 255 – 261 (1988).

[HNS 92] K. Hinrichs, J. Nievergelt, and P. Schorn, An all-round sweep algorithm for 2-dimensional nearest-neighbor problems, *Acta Informatica 29*(4), 1992.

[HS 86] S. Hart and M. Sharir, Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes, *Combinatorica 6*, 151 – 177 (1986).

[Hof 89] C. M. Hoffmann, *Geometric and Solid Modeling: An Introduction*, Morgan Kaufmann Publishers, San Mateo, CA, 1989.

[O'R 87] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, New York, 1987.

[PS 85] F. P. Preparata and M. I. Shamos, *Computational Geometry,* 3rd pr., Springer-Verlag, New York, 1985.

[SH 75] M. I. Shamos and D. Hoey, Closest-point problems, *Proceedings of the 16th Annual IEEE Symposium on Foundations of Computer Science*, 151 – 162 (1975).

[SH 76] M. I. Shamos and D. Hoey, Geometric intersection problems, *Proceedings of the 17th Annual IEEE Symposium on Foundations of Computer Science*, 208 – 215 (1976).