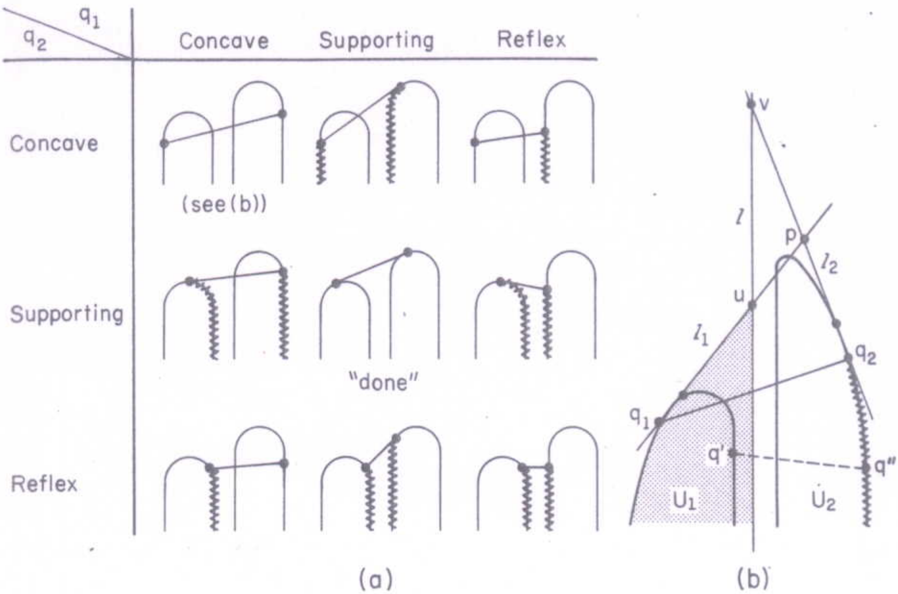

Figure 3.21 (a) All possible cases arising when choosing a vertex in U_1 and a vertex in U_2. (b) Illustration of the (concave, concave) case.

i. A pointer to a concatenable queue $Q[v]$, storing the portion of $U(v)$ not belonging to $U(\text{FATHER}[v])$ (if v is the root, then $Q[v] = U(v)$).
ii. An integer $J[v]$ denoting the position of the left support point on $U(v)$.

This interesting data structure uses only $O(N)$ space, where N is the size of the current point set. Indeed, the skeletal tree T has N leaves and $N-1$ internal nodes, while the points stored in the concatenable queues represent a partition of the point set.

Since the operations of splitting and splicing concatenable queues are standard, we shall concentrate on the operation BRIDGE for which Overmars and van Leeuwen (1981) propose the following solution.

Lemma 3.1. *The bridging of two separated convex chains of N points (in total) can be done in* $O(\log N)$ *steps.*

Proof. Given two U-hulls U_1 and U_2 and two vertices $q_1 \in U_1$ and $q_2 \in U_2$, each of these two vertices can be readily classified with respect to the segment $\overline{q_1 q_2}$ as either reflex, or supporting, or concave. (See Section 3.3.6 for an explanation of these terms.) Depending upon this classification there are nine possible cases, which are schematically illustrated in Figure 3.21(a). The wiggly subchains are those which can be eliminated from further contention for containing a support point. All cases are self-explanatory, except the case

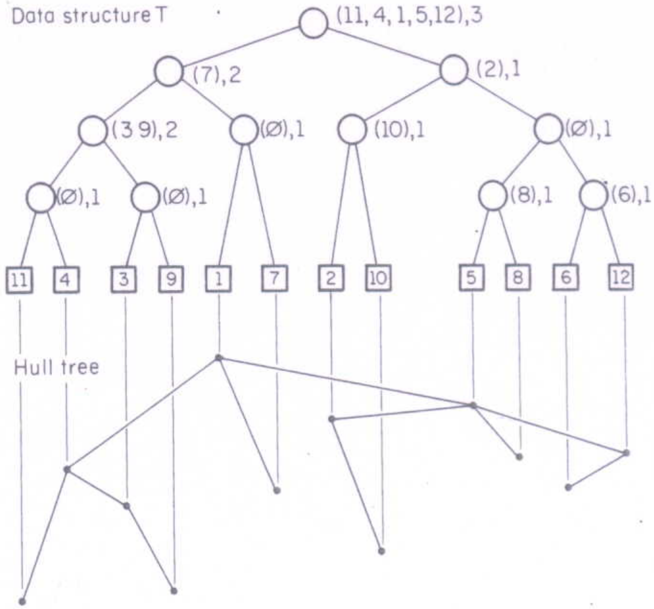

Figure 3.22 A planar point set and the corresponding data structure T.

$(q_1, q_2) =$ (concave, concave), which is further illustrated in Figure 3.21(b). Let line l_1 contain q_1 and its right neighbor on U_1; similarly let l_2 contain q_2 and its left neighbor on U_2, and let p be the intersection of l_1 and l_2. Recall that U_1 and U_2 are separated, by hypothesis, by a vertical line l. Assume, at first that p is to the right of l. We observe that support point p_1 can only belong to the shaded region, and that u has a lower ordinate than v. This implies that each vertex q'' on the subchain to the right of q_2 appears concave with respect to the segment $\overline{q'q''}$, where q' is any vertex of U_1. This shows that the chain to the right of q_2 can be eliminated from contention, but no similar statement can be made for the chain to the left of q_1. If intersection p is to the left of l, then we can show that the chain to the left of q_1 can be eliminated.

In all cases a portion of one or both hulls is eliminated. If this process starts from the roots of both trees representing U_1 and U_2, respectively, since these trees are balanced, BRIDGE(U_1, U_2) will run in time $O(\log N)$, where N is as usual the total number of vertices in the two hulls. □

With the function BRIDGE at our disposal, we may analyze the dynamic maintenance of the planar convex hull. A typical situation is illustrated in Figure 3.22. Here points are indexed by their order of insertion into the set. In the data structure T, each leaf corresponds to a point, whereas each nonleaf node corresponds to a bridge and is shown with the pair $Q[v]$, $J[v]$. It is immediate to realize that data structure T describes a free tree on the set of