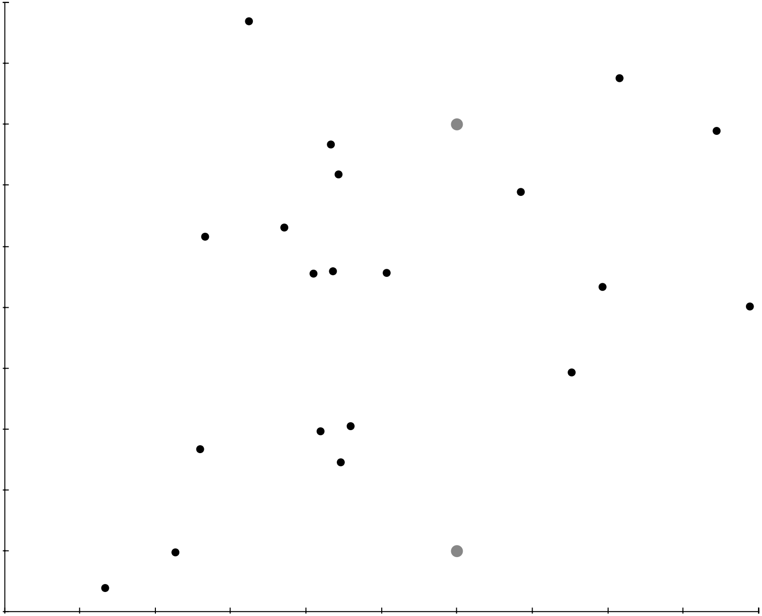
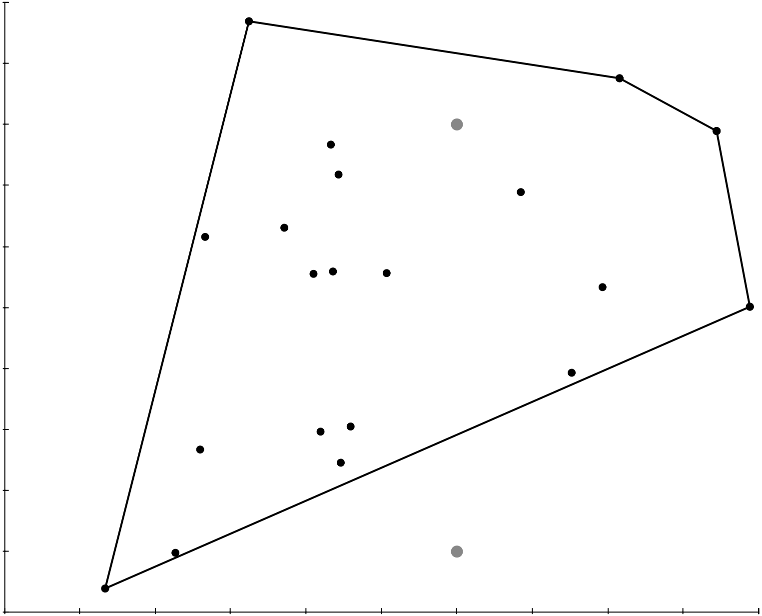
# Chapter 8: Planar Convex Hull

Elements of Parallel Computing

Eric Aubanel

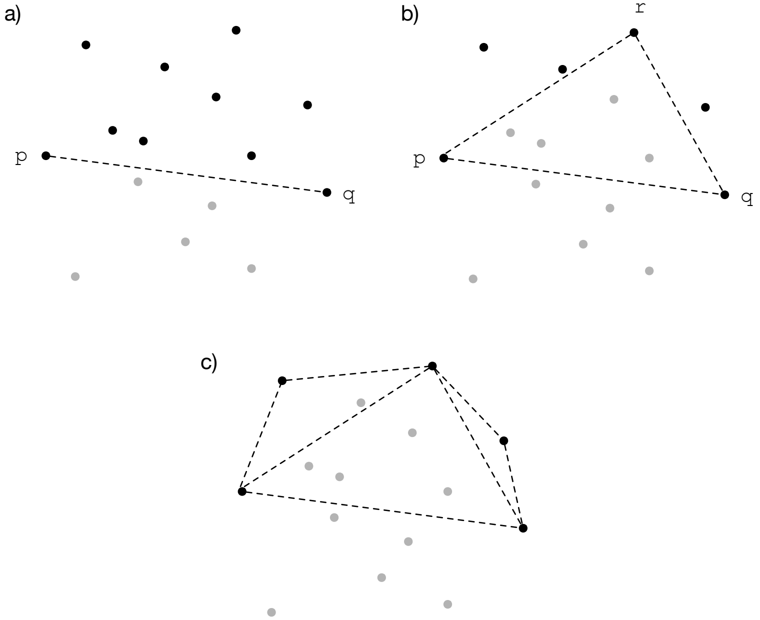




# Planar Convex Hull

The convex hull of a set of points $S$ on the plane is the smallest convex subset $H \subseteq S$ that contains all the points. $H$ is convex if a line through any pair of points in $H$ lies within the polygon formed by $H$.

# QuickHull



**Input**: Set $S$ of $n$ points on the plane.
**Output**: Set $H$ of points on convex hull.

$a \leftarrow$ `minIndex(`$\{c_x : c \in S\}$`)// index of point with minimum` $x$ `value`
$b \leftarrow$ `maxIndex(`$\{c_x : c \in S\}$`)`
$p \leftarrow S[a]$
$q \leftarrow S[b]$
$S_1 \leftarrow \{s \in S \mid s \text{ above } \overline{pq}\}$
$S_2 \leftarrow \{s \in S \mid s \text{ below } \overline{pq}\}$
$H_1 \leftarrow$ `subHull(`$S_1, p, q$`)// upper hull`
$H_2 \leftarrow$ `subHull(`$S_2, q, p$`)// lower hull`
$H = H1 \cup H2$

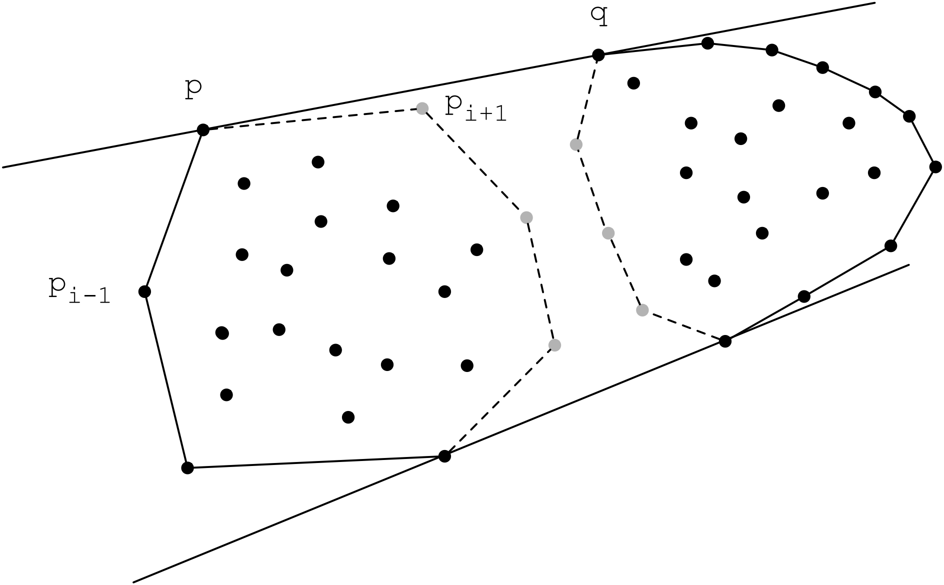
**Procedure** `subHull`($S$, $p$, $q$)
    **if** $|S| = 0$ **then**
        **return** $\{p\}$
    **end**
    $d \leftarrow$ `maxIndex`($\{$*distance between* $c$ *and* $\overline{pq} : c \in S\}$)
    $r \leftarrow S[d]$
    $S' \leftarrow \{s \in S \mid \text{points not in triangle } \overline{pqr}\}$
    $S_1 \leftarrow \{s \in S' \mid s \text{ closer to } \overline{pr}\}$
    $S_2 \leftarrow \{s \in S' \mid s \text{ closer to } \overline{rq}\}$
    $H_1 \leftarrow$ `subHull`($S_1$, $p$, $r$)
    $H_2 \leftarrow$ `subHull`($S_2$, $r$, $q$)
    **return** $H_1 \cup H_2$
**end**

# MergeHull

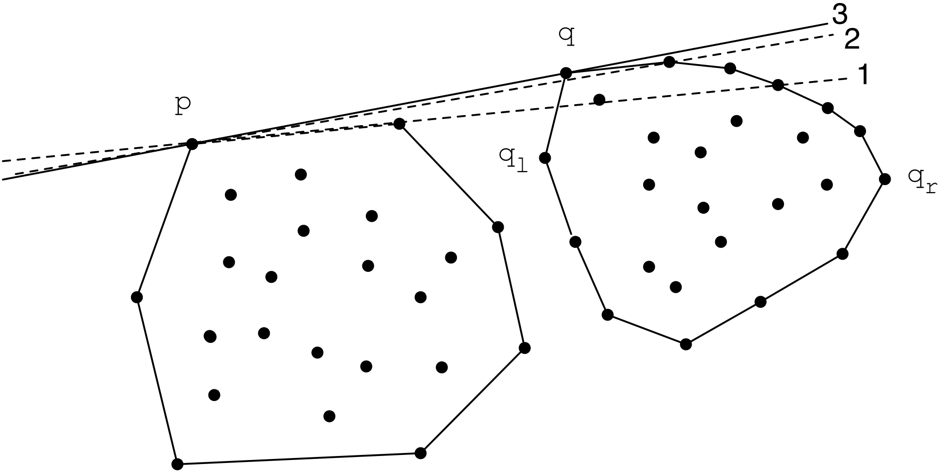
**Input**: Set $S$ of $n$ points on the plane, sorted in lexicographic order.
**Output**: Set $H$ of points on convex hull, in clockwise order.

**Procedure** `mergeHull`($S$)
    **if** $|S| \leq 3$ **then**
        **return** $S$
    **else**
        $H_1 \leftarrow$`mergeHull`$(S[0..|S|/2))$
        $H_2 \leftarrow$`mergeHull`$(S[|S|/2..|S|))$
        **return** `joinHulls`($H_1$, $H_2$)
    **end**
**end**

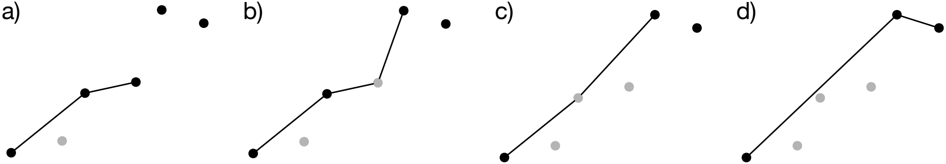
# MergeHull



# Finding Upper Tangent



# Graham Scan



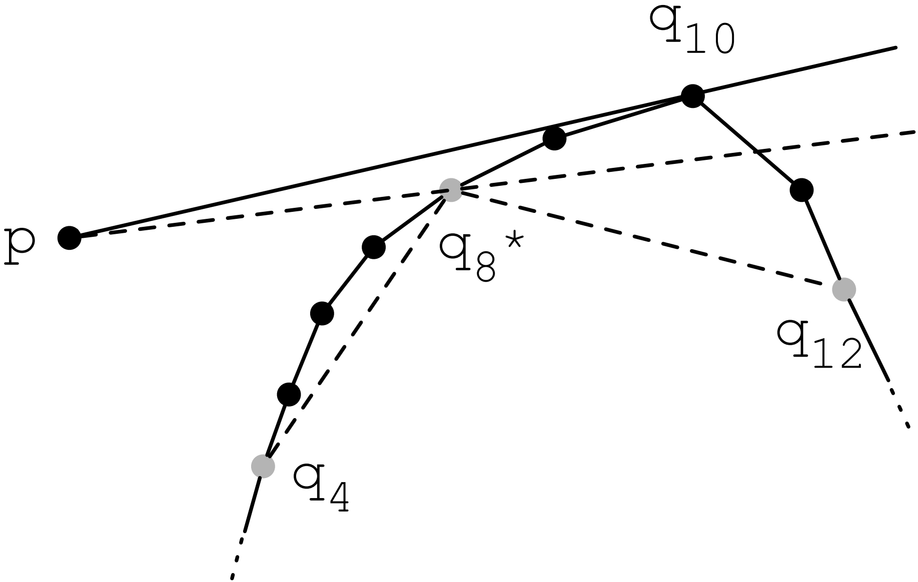
**Input**: Set $S$ of $n$ points on the plane, sorted in lexicographic order.
**Output**: Set $H$ of points on upper convex hull, in clockwise order.

add $S[0]$ and $S[1]$ to $H$
**for** $i \leftarrow 2$ *to* $n-1$ **do**
    add $S[i]$ to $H$
    **while** $|H| > 2$ *and last 3 points in H don't make a right hand turn* **do**
        remove second to last point of $H$
    **end**
**end**

# Parallel Design Exploration

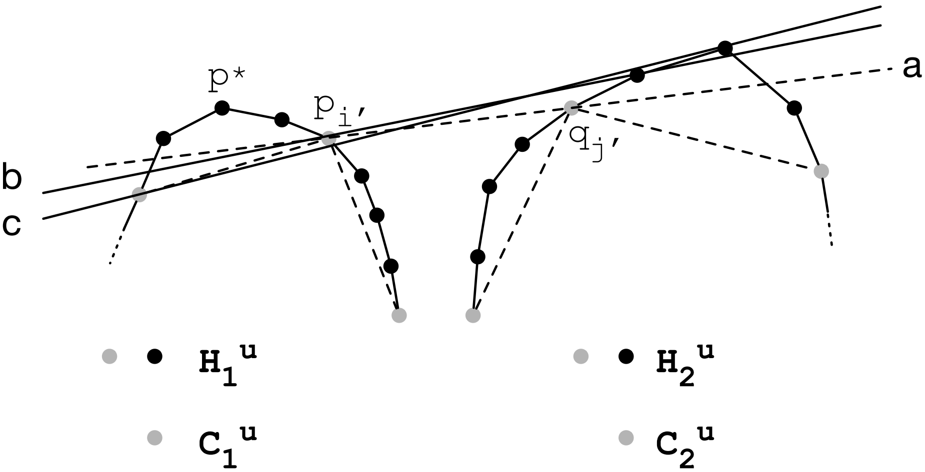
- QuickHull and MergeHull: divide-and-conquer
- $O(n \log n)$ work and $O(\log^2 n)$ depth
- QuickHull: $O(\log n)$ levels of $O(\log n)$ scan
- MergeHull: $O(\log n)$ levels of $O(\log n)$ computation of tangents
- Can reduce computation of tangent in MergeHull to $O(1)$ depth, by doing it in parallel

# Tangent between Point and Upper Hull



# Tangent between Upper Hulls

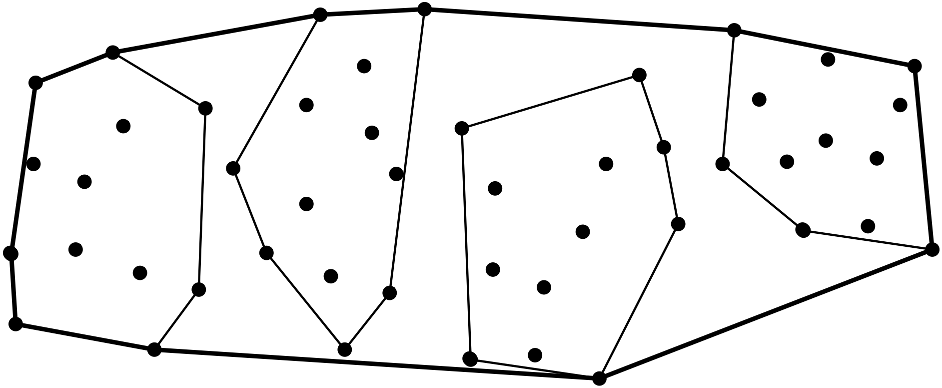


# Shared Memory MergeHull

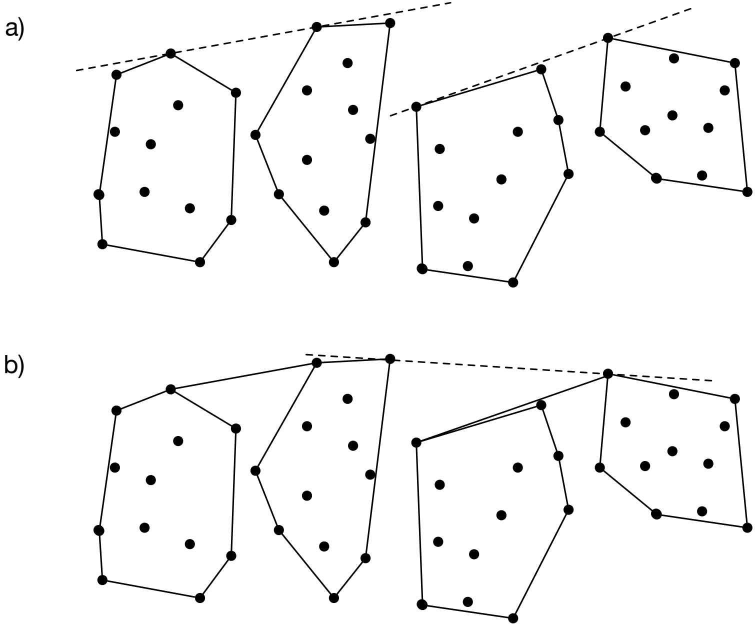
**Input**: Set $S$ of $n$ points on the plane, sorted in lexicographic order.
**Output**: Set $H$ of points on convex hull.
**Data**: $nt$ threads, with $id \in [0..nt)$.
**Data**: Size of $nt$ hulls given by array $N$.

**shared** S, H, N
$istart \leftarrow \lfloor id * n/nt \rfloor$
$iend \leftarrow \lfloor (id + 1) * n/nt \rfloor - 1$
`grahamScan(`$S$, $H$, $N$, $istart$, $iend$`)// each hull starts at` $H[istart]$
`hullMerge(`$H$, $N$`)`

# Merging 4 Hulls



# Recursive Merge



# All-Pairs Merge