2.3 QuickSort Demos

- Sedgewick 2-way partitioning
- Dijkstra 3-way partitioning
- Dijkstra 3-way partitioning
- Dual-pivot partitioning
Dual-pivot partitioning demo

Initialization.

- Choose $a[lo]$ and $a[hi]$ as partitioning items.
- Exchange if necessary to ensure $a[lo] \leq a[hi]$.

exchange $a[lo]$ and $a[hi]$
Dual-pivot partitioning demo

Initialization.

- Choose $a[lo]$ and $a[hi]$ as partitioning items.
- Exchange if necessary to ensure $a[lo] \leq a[hi]$. 
Dual-pivot partitioning demo

Main loop. Repeat until $i$ and $gt$ pointers cross.

- If $(a[i] < a[lo])$, exchange $a[i]$ with $a[lt]$ and increment $lt$ and $i$.
- Else if $(a[i] > a[hi])$, exchange $a[i]$ with $a[gt]$ and decrement $gt$.
- Else, increment $i$.

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$&lt; p_1$</th>
<th>$p_1 \leq \text{and} \leq p_2$</th>
<th>$?$</th>
<th>$&gt; p_2$</th>
<th>$p_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>lo</td>
<td>lt</td>
<td>$i$</td>
<td>$gt$</td>
<td>hi</td>
<td></td>
</tr>
</tbody>
</table>

exchange $a[i]$ and $a[lt]$; increment $lt$ and $i$
**Dual-pivot partitioning demo**

**Main loop.** Repeat until $i$ and $gt$ pointers cross.
- If $(a[i] < a[lo])$, exchange $a[i]$ with $a[lt]$ and increment $lt$ and $i$.
- Else if $(a[i] > a[hi])$, exchange $a[i]$ with $a[gt]$ and decrement $gt$.
- Else, increment $i$.

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$&lt; p_1$</th>
<th>$p_1 \leq$ and $\leq p_2$</th>
<th>?</th>
<th>$&gt; p_2$</th>
<th>$p_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑ lo</td>
<td>↑ lt</td>
<td>↑ i</td>
<td>↑ gt</td>
<td>↑ hi</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>K</th>
<th>E</th>
<th>A</th>
<th>Y</th>
<th>R</th>
<th>L</th>
<th>F</th>
<th>V</th>
<th>Z</th>
<th>Q</th>
<th>T</th>
<th>C</th>
<th>M</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>lo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>↑</td>
<td>lt</td>
<td>i</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>↑</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

exchange $a[i]$ and $a[lt]$; increment $lt$ and $i$
Main loop. Repeat until \(i\) and \(gt\) pointers cross.

- If \((a[i] < a[lo])\), exchange \(a[i]\) with \(a[lt]\) and increment \(lt\) and \(i\).
- Else if \((a[i] > a[hi])\), exchange \(a[i]\) with \(a[gt]\) and decrement \(gt\).
- Else, increment \(i\).

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
p_1 & < p_1 & p_1 \leq \text{ and } \leq p_2 & ? & > p_2 & p_2 \\
\hline
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
\text{lo} & \text{lt} & i & gt & \text{hi} & \\
\hline
\end{array}
\]

exchange \(a[i]\) and \(a[gt]\); decrement \(gt\)
Dual-pivot partitioning demo

Main loop. Repeat until $i$ and $gt$ pointers cross.

- If $(a[i] < a[lo])$, exchange $a[i]$ with $a[lt]$ and increment $lt$ and $i$.
- Else if $(a[i] > a[hi])$, exchange $a[i]$ with $a[gt]$ and decrement $gt$.
- Else, increment $i$.

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1$</td>
<td>$&lt; p_1$</td>
<td>$p_1 \leq \text{ and } \leq p_2$</td>
<td>$?$</td>
<td>$&gt; p_2$</td>
<td>$p_2$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\uparrow</td>
<td>\uparrow</td>
<td>\uparrow</td>
<td>\uparrow</td>
<td>\uparrow</td>
<td>\uparrow</td>
<td></td>
</tr>
<tr>
<td>\text{lo}</td>
<td>\text{lt}</td>
<td>\text{i}</td>
<td>\text{gt}</td>
<td>\text{hi}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

increment $i$
Dual-pivot partitioning demo

Main loop. Repeat until $i$ and $gt$ pointers cross.

- If $(a[i] < a[lo])$, exchange $a[i]$ with $a[lt]$ and increment $lt$ and $i$.
- Else if $(a[i] > a[hi])$, exchange $a[i]$ with $a[gt]$ and decrement $gt$.
- Else, increment $i$.

<table>
<thead>
<tr>
<th>p₁</th>
<th>&lt; p₁</th>
<th>p₁ ≤ and ≤ p₂</th>
<th>?</th>
<th>&gt; p₂</th>
<th>p₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>lo</td>
<td>lt</td>
<td>i</td>
<td>gt</td>
<td>hi</td>
<td></td>
</tr>
</tbody>
</table>

increment $i$
Dual-pivot partitioning demo

Main loop. Repeat until $i$ and $gt$ pointers cross.

- If $(a[i] < a[lo])$, exchange $a[i]$ with $a[lt]$ and increment $lt$ and $i$.
- Else if $(a[i] > a[hi])$, exchange $a[i]$ with $a[gt]$ and decrement $gt$.
- Else, increment $i$.

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$&lt; p_1$</th>
<th>$p_1 \leq \text{ and } \leq p_2$</th>
<th>$&lt;$</th>
<th>$&gt; p_2$</th>
<th>$p_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\uparrow$</td>
<td>$\uparrow$</td>
<td>$\uparrow$</td>
<td>$\uparrow$</td>
<td>$\uparrow$</td>
<td>$\uparrow$</td>
</tr>
<tr>
<td>$lo$</td>
<td>$lt$</td>
<td>$i$</td>
<td>$gt$</td>
<td>$hi$</td>
<td></td>
</tr>
</tbody>
</table>

increment $i$
**Main loop.** Repeat until i and gt pointers cross.

- If \((a[i] < a[lo])\), exchange \(a[i]\) with \(a[lt]\) and increment \(lt\) and \(i\).
- Else if \((a[i] > a[hi])\), exchange \(a[i]\) with \(a[gt]\) and decrement \(gt\).
- Else, increment \(i\).

<table>
<thead>
<tr>
<th>(p_1)</th>
<th>(&lt; p_1)</th>
<th>(p_1 \leq \text{ and } \leq p_2)</th>
<th>?</th>
<th>(&gt; p_2)</th>
<th>(p_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>lo</td>
<td>lt</td>
<td>i</td>
<td>gt</td>
<td>hi</td>
<td></td>
</tr>
</tbody>
</table>

exchange \(a[i]\) and \(a[lt]\); increment \(lt\) and \(i\)
Main loop. Repeat until i and gt pointers cross.
- If (a[i] < a[lo]), exchange a[i] with a[lt] and increment lt and i.
- Else if (a[i] > a[hi]), exchange a[i] with a[gt] and decrement gt.
- Else, increment i.

\[a[lt] \leftarrow a[i]\] and increment lt

\[a[i] \leftarrow a[gt]\] and decrement gt

**Tips:**
- The algorithm maintains two pointers, lo and hi, which are used to partition the array.
- The algorithm iterates until lo and hi pointers cross.
- The middle element is used as a pivot.
- If the element is less than the pivot, it is swapped with the left pointer.
- If the element is greater than the pivot, it is swapped with the right pointer.
- The algorithm continues until all elements are correctly positioned.

**Complexity:**
The time complexity of the dual-pivot quicksort is \(O(n \log n)\) in the best and average cases, and \(O(n^2)\) in the worst case.
Dual-pivot partitioning demo

Main loop. Repeat until $i$ and $gt$ pointers cross.

- If $(a[i] < a[lo])$, exchange $a[i]$ with $a[lt]$ and increment $lt$ and $i$.
- Else if $(a[i] > a[hi])$, exchange $a[i]$ with $a[gt]$ and decrement $gt$.
- Else, increment $i$.

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$&lt; p_1$</th>
<th>$p_1 \leq$ and $\leq p_2$</th>
<th>$&gt;$ $p_2$</th>
<th>$p_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>lo</td>
<td>lt</td>
<td>$i$</td>
<td>gt</td>
<td>hi</td>
</tr>
</tbody>
</table>

exchange $a[i]$ and $a[lt]$; increment $lt$ and $i$
Main loop. Repeat until \( i \) and \( gt \) pointers cross.

- If \( (a[i] < a[lo]) \), exchange \( a[i] \) with \( a[lt] \) and increment \( lt \) and \( i \).
- Else if \( (a[i] > a[hi]) \), exchange \( a[i] \) with \( a[gt] \) and decrement \( gt \).
- Else, increment \( i \).

\[
\begin{array}{ccccccc}
| p_1 | < p_1 | p_1 \leq \text{and} \leq p_2 | ? | > p_2 | p_2 \\
\hline
\uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow \\
lo | lt | i | gt | hi \\
\end{array}
\]

\text{exchange } a[i] \text{ and } a[gt]; \text{ decrement } gt
Dual-pivot partitioning demo

Main loop. Repeat until i and gt pointers cross.

- If (a[i] < a[lo]), exchange a[i] with a[l] and increment l and i.
- Else if (a[i] > a[hi]), exchange a[i] with a[gt] and decrement gt.
- Else, increment i.

---

**Table:**

<table>
<thead>
<tr>
<th>p1</th>
<th>&lt; p1</th>
<th>p1 ≤ and ≤ p2</th>
<th>?</th>
<th>&gt; p2</th>
<th>p2</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>lo</td>
<td>lt</td>
<td>i</td>
<td></td>
<td>gt</td>
<td>hi</td>
</tr>
</tbody>
</table>

---

exchange a[i] and a[gt]; decrement gt
**Dual-pivot partitioning demo**

**Main loop.** Repeat until i and gt pointers cross.

- If $(a[i] < a[lo])$, exchange $a[i]$ with $a[lt]$ and increment $lt$ and $i$.
- Else if $(a[i] > a[hi])$, exchange $a[i]$ with $a[gt]$ and decrement $gt$.
- Else, increment $i$.

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$&lt; p_1$</th>
<th>$p_1 \leq$ and $\leq p_2$</th>
<th>$?$</th>
<th>$&gt; p_2$</th>
<th>$p_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>lo</td>
<td>lt</td>
<td>i</td>
<td>gt</td>
<td>hi</td>
<td></td>
</tr>
</tbody>
</table>

Increment $i$
Dual-pivot partitioning demo

Main loop. Repeat until i and gt pointers cross.

- If \( a[i] < a[lo] \), exchange \( a[i] \) with \( a[lt] \) and increment \( lt \) and \( i \).
- Else if \( a[i] > a[hi] \), exchange \( a[i] \) with \( a[gt] \) and decrement \( gt \).
- Else, increment \( i \).

```
<table>
<thead>
<tr>
<th>p1</th>
<th>&lt; p1</th>
<th>p1 ≤ and ≤ p2</th>
<th>?</th>
<th>&gt; p2</th>
<th>p2</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>lo</td>
<td>lt</td>
<td>i</td>
<td>gt</td>
<td>hi</td>
<td></td>
</tr>
</tbody>
</table>
```

stop when pointers cross
Dual-pivot partitioning demo

Finalize.

- Exchange \(a[lo]\) with \(a[-lt]\).
- Exchange \(a[hi]\) with \(a[+gt]\).
Dual-pivot partitioning demo

Finalize.
- Exchange \( a[lo] \) with \( a[--lt] \).
- Exchange \( a[hi] \) with \( a[+gt] \).

<table>
<thead>
<tr>
<th>( p_1 &lt; p_1 )</th>
<th>( p_1 )</th>
<th>( p_1 \leq \text{ and } \leq p_2 )</th>
<th>( p_2 )</th>
<th>( &gt; p_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \uparrow )</td>
<td>( \uparrow )</td>
<td>( \uparrow )</td>
<td>( \uparrow )</td>
<td>( \uparrow )</td>
</tr>
<tr>
<td>( \text{lo} )</td>
<td>( \text{lt} )</td>
<td>( \text{gt} )</td>
<td>( \text{hi} )</td>
<td></td>
</tr>
</tbody>
</table>

3-way partitioned