Extra Lecture – Introduction to Sorting

- What is sorting?

- Given an array $R[1..N]$ of $N$ numbers, re-order the elements of $R[]$ such that after re-ordering,
  - $R[1] \leq R[2] \leq \ldots \leq R[N]$

- Sorting is an extremely well studied problem in Computing. Many algorithms exist, including
  ➤ bubble sort
  ➤ heap sort
  ➤ quick sort

Bubble Sort [Knuth, Vol 3, p.107]

Algorithm B (BubbleSort)

B1. [Initialize BOUND] Set BOUND := $N$

B2. [Loop on $j$] Set $t := 0$. Perform B3 for $j=1,2,\ldots,\text{BOUND}-1$ and then go to step B4.


B4. [Any exchanges?] If $t=0$, the algorithm terminates. Otherwise set BOUND := $t$ and return to step B2.

BubbleSort - Notes

- BOUND is the highest index for which the element is not known to be in its final (sorted) position.
  - $\text{BOUND} = N \Rightarrow$ Nothing is known about ordering
  - $\text{BOUND} = 0 \Rightarrow$ Array is in perfect order

- $t$ holds the last index at which an exchange was performed
  - $t=0 \Rightarrow$ No exchanges were performed

- Algorithm as presented sorts in ascending order of $R_j$

Bubble Sort - Example

<table>
<thead>
<tr>
<th>$R[1]$</th>
<th>43</th>
<th>34</th>
<th>34</th>
<th>34</th>
<th>34</th>
<th>34</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R[2]$</td>
<td>34</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>$R[3]$</td>
<td>64</td>
<td>64</td>
<td>48</td>
<td>35</td>
<td>43</td>
<td>43</td>
</tr>
<tr>
<td>$R[4]$</td>
<td>64</td>
<td>48</td>
<td>35</td>
<td>48</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>$R[5]$</td>
<td>48</td>
<td>35</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>$R[6]$</td>
<td>35</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>$R[7]$</td>
<td>84</td>
<td>84</td>
<td>84</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
</tbody>
</table>