CS61B Lectures #28

Today:

• Lower bounds on sorting by comparison
• Distribution counting, radix sorts

Readings: Today: DS(IJ), Chapter 8; Next topic: Chapter 9.
Better than $N \lg N$?

- Can prove that *if all you can do to keys is compare them*, then sorting must take $\Omega(N \lg N)$.

- Basic idea: there are $N!$ possible ways the input data could be scrambled.

- Therefore, your program must be prepared to do $N!$ different combinations of data-moving operations.

- Therefore, there must be $N!$ possible combinations of outcomes of all the if-tests in your program, since those determine what move gets moved where (we’re assuming that comparisons are 2-way).

```
Decision Tree
Height $\propto$ Sorting time
```

```
T
  a < b
    b < c
      abc
      acb
    a < c
      acb
      cab
  F
    a < c
      bac
      bca
    b < c
      bca
      cba
```
Necessary Choices

• Since each if-test goes two ways, number of possible different outcomes for $k$ if-tests is $2^k$.

• Thus, need enough tests so that $2^k \geq N!$, which means $k \geq \lg N!$.

• Using Stirling’s approximation,

$$N! \in \sqrt{2\pi N} \left(\frac{N}{e}\right)^N \left(1 + \Theta\left(\frac{1}{N}\right)\right),$$

$$\lg(N!) \in \frac{1}{2} (\lg 2\pi + \lg N) + N \lg N - N \lg e + \lg \left(1 + \Theta\left(\frac{1}{N}\right)\right)$$

$$= \Theta(N \lg N)$$

• This tells us that $k$, the worst-case number of tests needed to sort $N$ items by comparison sorting, is in $\Omega(N \lg N)$: there must be cases where we need (some multiple of) $N \lg N$ comparisons to sort $N$ things.
Beyond Comparison: Distribution

• But suppose we can do more than compare keys?
• For example, how can we sort a set of $N$ different integer keys whose values range from 0 to $kN$, for some small constant $k$?
• One technique is distribution sorting:
  - Put the integers into $N$ buckets; integer $p$ goes to bucket $\lfloor p/k \rfloor$.
  - At most $k$ keys per bucket, so catenate and use insertion sort, which will now be fast.
• E.g., $k = 2$, $N = 10$:

Start:

14 3 10 13 4 2 19 17 0 9

In buckets:

| 0 | 3 2 | 4 | | 9 | 10 | 13 | 14 | 17 | 19 |

• Now insertion sort is fast. Putting the data in buckets takes time $\Theta(N)$, and insertion sort takes $\Theta(kN)$. When $k$ is fixed (constant), we have sorting in time $\Theta(N)$. 
Distribution Counting

- Another technique: **count** the number of items $< 1$, $< 2$, etc.
- If $M_p = \#\text{items with value } < p$, then in sorted order, the $j^\text{th}$ item with value $p$ must be item $\#M_p + j$.
- Suppose that one has a set of numbers in the range $[0, 1000)$ (possibly with duplicates) and that exactly 15 of them are less than 50, which is also in the set. Then the result of sorting will look like this:

```
result:    0 50 999
   15
```

- In other words, the count of numbers $< k$ gives the index of $k$ in the output array.
- If there are $N$ items in the range $0..M-1$, this gives another **linear-time**—$\Theta(M + N)$—algorithm (We include $M$ and $N$ here to allow for both duplicates and for cases where $M \gg N$.)
- [Postscript on notation: the notations $[A, B]$, $(A, B)$, $[A, B)$, and $(A, B]$ above refer to **intervals**. The use of parentheses vs. square brackets reflects the distinction between open and closed intervals. Thus $x \in [A, B]$ iff $A \leq x \leq B$, while $x \in (A, B)$ iff $A \leq x < B$, etc.]
Distribution Counting Example

- Suppose all items are between 0 and 9 as in this example:

```
  7 0 4 0 9 1 9 1 9 5 3 7 3 1 6 7 4 2 0
```

<table>
<thead>
<tr>
<th>Counts</th>
<th>Running sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 3 1 2 2 1 1 3 0 3</td>
<td>&lt; 0 &lt; 1 &lt; 2 &lt; 3 &lt; 4 &lt; 5 &lt; 6 &lt; 7 &lt; 8 &lt; 9</td>
</tr>
<tr>
<td>0 1 2 3 4 5 6 7 8 9</td>
<td>0 0 0 1 1 1 2 3 3 4 4 5 6 7 7 7 9 9 9</td>
</tr>
<tr>
<td></td>
<td>0 3 6 9 11 12 13 16 16</td>
</tr>
</tbody>
</table>

- “Counts” line gives # occurrences of each key.
- “Running sum” gives cumulative count of keys < each value...
- ...which tells us where to put each key:
- The first instance of key $k$ goes into slot $m$, where $m$ is the number of key instances that are < $k$; next $k$ goes into slot $m + 1$, etc.
Distribution Counting Example (II)

Counts

<table>
<thead>
<tr>
<th>3</th>
<th>3</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>3</th>
<th>0</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Running sum of Counts

<table>
<thead>
<tr>
<th>0</th>
<th>3</th>
<th>6</th>
<th>7</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Next positions

<table>
<thead>
<tr>
<th>0</th>
<th>3</th>
<th>6</th>
<th>7</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Output

<table>
<thead>
<tr>
<th>0</th>
<th>3</th>
<th>6</th>
<th>9</th>
<th>12</th>
<th>15</th>
<th>18</th>
</tr>
</thead>
</table>
Distribution Counting Example (II)

Counts

Running sum of Counts

Next positions

Output
### Distribution Counting Example (II)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Counts</strong></td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>Running sum of Counts</strong></td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td><strong>Next positions</strong></td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

|    | 0 | 3 | 6 | 7 | 9 | 12 | 15 | 18 |
|----|---|---|---|---|---|-----|-----|
| **Output** | 0 | 3 | 6 | 9 | 12 | 15 | 18 |  |  |  |  |  |  |  |  |  |  |  |  |  |

Last modified: Thu Oct 28 22:01:29 2021
Distribution Counting Example (II)

Counts

0 3 3 1 2 2 1 1 3 0 3

Running sum of Counts

0 1 3 6 7 9 11 12 13 16 16

Next positions

0 1 3 6 7 10 11 12 14 16 16

Output

0 3 6 9 12 15 18
## Distribution Counting Example (II)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Counts</strong></td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>Running sum of Counts</strong></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td><strong>Next positions</strong></td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Last modified: Thu Oct 28 22:01:29 2021
Distribution Counting Example (II)

Counts

0 1 2 3 4 5 6 7 8 9

Running sum of Counts

0 1 2 3 4 5 6 7 8 9

Next positions

0 1 2 3 4 5 6 7 8 9

Output

0 0 0 0 0 0 0 0

0 3 6 9 12 15 18
## Distribution Counting Example (II)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counts</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Running sum of Counts</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Next positions</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Output</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td>7</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Last modified: Thu Oct 28 22:01:29 2021
Distribution Counting Example (II)

Counts

<table>
<thead>
<tr>
<th>3</th>
<th>3</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>3</th>
<th>0</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Running sum of Counts

<table>
<thead>
<tr>
<th>0</th>
<th>3</th>
<th>6</th>
<th>7</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Next positions

<table>
<thead>
<tr>
<th>2</th>
<th>4</th>
<th>6</th>
<th>7</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Output

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
<th>4</th>
<th>7</th>
<th>9</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
</tr>
</tbody>
</table>
Distribution Counting Example (II)

Counts

| 7 | 0 | 4 | 0 | 9 | 1 | 9 | 1 | 9 | 5 | 3 | 7 | 3 | 1 | 6 | 7 | 4 | 2 | 0 |

Running sum of Counts

<table>
<thead>
<tr>
<th>0</th>
<th>3</th>
<th>3</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>3</th>
<th>0</th>
<th>3</th>
</tr>
</thead>
</table>

Next positions

<table>
<thead>
<tr>
<th>0</th>
<th>6</th>
<th>7</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
</table>

Output

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>4</th>
<th>7</th>
<th>9</th>
<th>9</th>
</tr>
</thead>
</table>

Output positions:

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>4</th>
<th>7</th>
<th>9</th>
<th>9</th>
</tr>
</thead>
</table>

Last modified: Thu Oct 28 22:01:29 2021
### Distribution Counting Example (II)

#### Counts

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

#### Running sum of Counts

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

#### Next positions

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

#### Output

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>4</th>
<th>7</th>
<th>9</th>
<th>9</th>
<th>9</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

---

Last modified: Thu Oct 28 22:01:29 2021
Distribution Counting Example (II)

Counts

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Running sum of Counts

<table>
<thead>
<tr>
<th>0</th>
<th>3</th>
<th>6</th>
<th>7</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Next positions

<table>
<thead>
<tr>
<th>2</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>10</th>
<th>12</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Output

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>9</th>
<th>9</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Distribution Counting Example (II)

| 7 | 0 | 4 | 0 | 9 | 1 | 9 | 1 | 9 | 5 | 3 | 7 | 3 | 1 | 6 | 7 | 4 | 2 | 0 |

<table>
<thead>
<tr>
<th>3</th>
<th>3</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>3</th>
<th>0</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>3</th>
<th>6</th>
<th>7</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>5</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>9</th>
<th>9</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td>19</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
</tbody>
</table>

**Counts**

**Running sum of Counts**

**Next positions**

**Output**
Distribution Counting Example (II)

Counts

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Running sum of Counts

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Next positions

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Output

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

Last modified: Thu Oct 28 22:01:29 2021
**Distribution Counting Example (II)**

| 7 | 0 | 4 | 0 | 9 | 1 | 9 | 1 | 9 | 5 | 3 | 7 | 3 | 1 | 6 | 7 | 4 | 2 | 0 |

<table>
<thead>
<tr>
<th>3</th>
<th>3</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>3</th>
<th>0</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>3</th>
<th>6</th>
<th>7</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>5</th>
<th>6</th>
<th>9</th>
<th>10</th>
<th>12</th>
<th>12</th>
<th>15</th>
<th>16</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>3</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>7</th>
<th>9</th>
<th>9</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Distribution Counting Example (II)

Counts

| 7 | 0 | 4 | 0 | 9 | 1 | 9 | 1 | 9 | 5 | 3 | 7 | 3 | 1 | 6 | 7 | 4 | 2 | 0 |

Running sum of Counts

| 0 | 3 | 3 | 6 | 7 | 9 | 11 | 12 | 13 | 16 | 16 |

Next positions

| 2 | 6 | 6 | 9 | 10 | 12 | 12 | 15 | 16 | 19 |

Output

| 0 | 0 | 1 | 1 | 1 | 3 | 3 | 4 | 5 | 7 | 7 | 9 | 9 | 9 | 12 | 15 | 18 |

Last modified: Thu Oct 28 22:01:29 2021
Distribution Counting Example (II)

Counts

Running sum of Counts

Next positions

Output
### Distribution Counting Example (II)

<table>
<thead>
<tr>
<th>Counts</th>
<th>Running sum of Counts</th>
<th>Next positions</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 0 4 0 9 1 9 1 9 5 3 7 3 1 6</td>
<td>0 3 6 7 9 11 12 16 16</td>
<td>2 6 6 9 10 12 13 16 16</td>
<td>0 0 1 1 1 3 3 4 5 6 7 7 7 9 9 9</td>
</tr>
</tbody>
</table>

Last modified: Thu Oct 28 22:01:29 2021

CS61B: Lecture #28  23
Distribution Counting Example (II)

| 7 | 0 | 4 | 0 | 9 | 1 | 9 | 1 | 9 | 5 | 3 | 7 | 3 | 1 | 6 | 7 | 4 | 2 | 0 |

<table>
<thead>
<tr>
<th>3</th>
<th>3</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>3</th>
<th>0</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>3</th>
<th>6</th>
<th>7</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>6</th>
<th>6</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>16</th>
<th>16</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

| 0 | 0 | 1 | 1 | 1 | 3 | 3 | 4 | 4 | 5 | 6 | 7 | 7 | 7 | 9 | 9 | 9 | 9 |
| 0 | 3 | 6 | 9 | 12 | 15 | 18 |
Distribution Counting Example (II)

**Counts**

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

**Running sum of Counts**

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

**Next positions**

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>16</td>
<td>16</td>
<td>19</td>
</tr>
</tbody>
</table>

**Output**

| 0 | 0 | 1 | 1 | 1 | 2 | 3 | 3 | 4 | 4 | 5 | 6 | 7 | 7 | 7 | 9 | 9 | 9 | 9 |
| 0 | 3 | 6 | 9 | 12 | 15 | 18 | | | | | | | | | | | | | | |
### Distribution Counting Example (II)

#### Counts

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

#### Running sum of Counts

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

#### Next positions

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

#### Output

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Last modified: Thu Oct 28 22:01:29 2021
Radix Sort

Idea: Sort keys *one character at a time*.

- Can use distribution counting for each digit.
- Can work either right to left (LSD radix sort) or left to right (MSD radix sort)
- LSD radix sort is venerable: used for punched cards. Example:

  Initial: set, cat, cad, con, bat, can, be, let, bet

  Pass 1
  (by char #2)
  be, cad, con, set, bet, let, cat, can

  Pass 2
  (by char #1)
  bat, bet, con, cat, let, set

  Pass 3
  (by char #0)
  bat, be, cad, can, con, let, set
MSD Radix Sort

- A bit more complicated: must keep lists from each step separate
- But, can stop processing 1-element lists

<table>
<thead>
<tr>
<th></th>
<th>posn</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
</tr>
<tr>
<td>* set, cat, cad, con, bat, can, be, let, bet</td>
<td>0</td>
</tr>
<tr>
<td>* bat, be, bet / cat, cad, con, can / let / set</td>
<td>1</td>
</tr>
<tr>
<td>bat / * be, bet / cat, cad, con, can / let / set</td>
<td>2</td>
</tr>
<tr>
<td>bat / be / bet / * cat, cad, con / can / let / set</td>
<td>1</td>
</tr>
<tr>
<td>bat / be / bet / * cat, cad, can / con / let / set</td>
<td>2</td>
</tr>
<tr>
<td>bat / be / bet / cad / can / cat / con / let / set</td>
<td></td>
</tr>
</tbody>
</table>

- Here, slashes divide partially sorted sublists that will never be moved into the space occupied by other sublists.
- Asterisks mark a sublist to be sorted on some character position.
Performance of Radix Sort

- Radix sort takes $\Theta(B)$ time where $B$ is total size of the key data.
- Have measured other sort times as functions of #records.
- How to compare?
  - To have $N$ different records, one must have keys at least $\Theta(lg N)$ long [why?]
  - Furthermore, comparison actually takes time $\Theta(K)$ where $K$ is size of key in worst case [why?]
- So $N \lg N$ comparisons really means $N(\lg N)^2$ operations.
- While radix sort would take $B = N \lg N$ time for $N$ records with minimal-length $(\lg N)$ keys.
- On the other hand, we must work to get good constant factors with radix sort.
And Don’t Forget Search Trees

Idea: A search tree is in sorted order, when read in inorder.

- Need balance to really use for sorting [next topic].
- Given balance, same performance as heapsort: $N$ insertions in time $\lg N$ each, plus $\Theta(N)$ to traverse, gives

$$\Theta(N + N \lg N) = \Theta(N \lg N)$$
Summary

• Insertion sort: $\Theta(Nk)$ comparisons and moves, where $k$ is maximum amount data is displaced from final position.
  - Good for small datasets or almost ordered data sets.

• Quicksort: $\Theta(N \lg N)$ with good constant factor if data is not pathological. Worst case $O(N^2)$.

• Merge sort: $\Theta(N \lg N)$ guaranteed. Good for external sorting.

• Heapsort, treesort with guaranteed balance: $\Theta(N \lg N)$ guaranteed.

• Radix sort, distribution sort: $\Theta(B)$ (number of bytes). Also good for external sorting.