Week 4 Tutorial

Ru Wang
CSE Dept, CUHK

Adapted from the slides of the previous offerings of the course
Outline

• Review recursion principle
• Review merge sort
• A variant of binary search
• A variant of merge sort
• Closest pair problem
Review – Recursion Principle

• When dealing with a subproblem (same problem but with a smaller input)

1. Consider it solved;

2. Use its output to design the rest of the algorithm.
Review – Merge Sort

• Identify the subproblems:
  • Sort the first half of the array $S$.
  • Sort the second half of $S$.

The original array $S$: 35 28 38 17 41 88 26 9

Subproblems: 35 28 38 17 41 88 26 9

Output: 17 28 35 38 9 26 41 88
Review - Merge Operation

• Merge 2 sorted arrays into a single sorted array
Review - Merge Operation

- Set $i, j$ to 1
- Compare 17 and 9
- 9 is smaller
- Place 9 into the new array and increase $j$ by 1
Review - Merge Operation

- Compare 17 and 26
- 17 is smaller
- Place 17 into the new array and increase $i$ by 1

<table>
<thead>
<tr>
<th>$i$</th>
<th>17</th>
<th>28</th>
<th>35</th>
<th>38</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
<td>9</td>
<td>26</td>
<td>41</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>17</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Review - Merge Operation

• Compare 28 and 26
• 26 is smaller
• Place 26 into the new array and increase $j$ by 1
Review - Merge Operation

• Continue the above process until we have placed all elements into the new array
• Single pass over all the input elements
• Time complexity: $O(n)$
Execution Trace - Merge Sort

| 35 | 28 | 38 | 17 | 41 | 88 | 26 | 9 |
Execution Trace - Merge Sort

35 28 38 17 41 88 26 9

35 28 38 17
Execution Trace - Merge Sort

35 28 38 17 41 88 26 9

35 28 38 17

35 28
Execution Trace - Merge Sort

35 28 38 17 41 88 26 9

35 28 38 17

35 28

35
Execution Trace - Merge Sort

35 28 38 17 41 88 26 9

35 28 38 17

35 28

35 28
Execution Trace - Merge Sort

35 28 38 17 41 88 26 9

35 28 38 17

28 35
Execution Trace - Merge Sort

35 28 38 17 41 88 26 9

35 28 38 17

28 35 38 17
Execution Trace - Merge Sort

```
35 28 38 17 41 88 26 9
```

```
35 28 38 17
```

```
28 35 38 17
```

```
38
```

```
38
```
Execution Trace - Merge Sort

```
35 28 38 17 41 88 26 9
```

```
35 28 38 17
```

```
28 35 38 17
```

```
38 17
```

```
38 17
```
Execution Trace - Merge Sort

```
35 28 38 17 41 88 26 9
```

```
35 28 38 17
```

```
28 35 17 38
```
Execution Trace - Merge Sort

35 28 38 17 41 88 26 9

17 28 35 38
Execution Trace - Merge Sort

```
 35 28 38 17 41 88 26 9
```

```
 17 28 35 38
```

```
 41 88 26 9
```

```
 41 88
```

```
 41
```
Execution Trace - Merge Sort

35 28 38 17 41 88 26 9

17 28 35 38

9 26 41 88
Execution Trace - Merge Sort

\[
\begin{array}{cccccccc}
9 & 17 & 26 & 28 & 35 & 38 & 41 & 88 \\
\end{array}
\]
A Variant of Binary Search

- Instead of comparing the target value with the middle element, we compare the target with the \( \lceil n/3 \rceil \)-th element each time.
Time Complexity

- In the worst case, after each comparison, two-thirds of the active elements are left.

- Solution
  - \( T(1) = O(1) \)
  - \( T(n) \leq T\left(\left\lfloor \frac{2n}{3} \right\rfloor \right) + O(1) \)
  - Solving the recurrence gives \( T(n) = O(\log n) \).
Time Complexity

• What if we compare the target with the $\left\lfloor \frac{n}{300} \right\rfloor$-th element?

• The time complexity is still $O(\log n)$!
  • Try verifying this by yourself.

• In general, if the comparison is made to the $\left\lfloor \frac{n}{k} \right\rfloor$-th element for some constant $k > 1$, the time complexity is still $O(\log n)$. 
Generalized Merge Operation

Merge 2 sorted arrays $A$ and $B$, of lengths $m$ and $n$.

• Set $i, j$ to 1.
• Compare 9 with 25.
• Place 9 into the new array and increase $i$ by 1.
Generalized Merge Operation

• Repeat the process until we have put all the elements of one input array into the new array.
Generalized Merge Operation

• Append the remaining elements to the new array.
• Time complexity: $O(m + n)$.  

\[
\begin{array}{c}
| i | & | j | \\
\hline
28 & 35 & 38 & 9 & 17 & 26 & 41 & 88 \\
\hline
\end{array}
\]

\[
\begin{array}{c}
| A | & | B | \\
\hline
9 & 17 & 28 & 35 & 38 & 39 & 41 & 88 \\
\hline
\end{array}
\]
A Variant of Merge Sort

• Solve the subproblems:
  • Sort the first \( \left\lfloor \frac{n}{3} \right\rfloor \) elements of the array \( S \).
  • Sort the rest of \( S \).

• Merge the 2 sorted arrays of different lengths.

The original array \( S \):

\[
\begin{array}{ccccccccc}
35 & 28 & 38 & 17 & 41 & 88 & 26 & 9 \\
\end{array}
\]

Subproblems:

\[
\begin{array}{ccc}
35 & 28 & 38 \\
17 & 41 & 88 & 26 & 9 \\
\end{array}
\]
Time Complexity

- The merging takes $O \left( \left\lfloor \frac{2n}{3} \right\rfloor + \left\lceil \frac{n}{3} \right\rceil \right) = O(n)$ time.
- Recurrence
  - $T(1) = O(1)$
  - $T(n) \leq T \left( \left\lfloor \frac{2n}{3} \right\rfloor \right) + T \left( \left\lceil \frac{n}{3} \right\rceil \right) + O(n)$
  - Solving the recurrence gives $T(n) = O(n \log n)$.
  - The recurrence can be solved with the substitution method (a regular exercise).
A Bonus Problem: Closest Pair

• Problem input:
  • Two **unsorted** sequences $A$ and $B$ with $m$ and $n$ integers
  • $n < m$

• Goal: Find a pair $(x, y)$, $x$ from $A$ and $y$ from $B$, with the minimum $|x - y|$.

<table>
<thead>
<tr>
<th>Sequence A</th>
<th>1</th>
<th>20</th>
<th>9</th>
<th>23</th>
<th>2</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence B</td>
<td>11</td>
<td>8</td>
<td>7</td>
<td>12</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>
A Bonus Problem: Closest Pair

• This problem can be solved in $O(m \log n)$ time.
  • Sort the shorter sequence.
  • Then, use elements of the longer sequence to perform binary searches.

• Note: $O(m \log n)$ is better than $O(m \log m)$ when $n << m$.

<table>
<thead>
<tr>
<th>Sequence A</th>
<th>1</th>
<th>20</th>
<th>9</th>
<th>23</th>
<th>2</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sorted Sequence B</td>
<td>7</td>
<td>8</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>