continuing - - -

\[ T(n) = c_1 n + c_2 (n-1) + c_3 (n-1) + c_4 \sum_{j=2}^{n} b_j \]

\[ + c_5 \sum_{j=2}^{n} (b_j - 1) + c_6 \sum_{j=2}^{n} (b_j - 1) + c_7 (n-1) \]

\[ = (c_4 + c_5 + c_6) \sum_{j=2}^{n} b_j + (c_1 + c_2 + c_3 - c_5 - c_6 + c_7) n \]

\[ + (-c_2 - c_3 + c_5 + c_6 - c_7) \]

**Best case:** \( b_j = 1 \) for \( 2 \leq j \leq n \).

\[ \therefore \sum_{j=2}^{n} b_j = n-1 \]

\[ T(n) = (c_1 + c_2 + c_3 + c_4 + c_7) n + (-c_2 - c_3 - c_4 - c_7) \]
worst case: \[ t_j = \frac{1}{n} \]

\[
\sum_{j=2}^{n} t_j = \sum_{i=1}^{n} i - 1 = \frac{n(n+1)}{2} - 1
\]

\[
-1(n) = \left( \frac{\sum_{i=4}^{i=6} i}{2} \right) n^2 + ( \quad ) n + ( \quad )
\]

avg. case: assume every input permutation is equally likely.

\[
t_j = \frac{1}{2}
\]

\[
\sum_{j=2}^{n} t_j = \frac{1}{2} \cdot \sum_{i=2}^{n} i = \frac{1}{2} \left( \frac{n(n+1)}{2} - 1 \right)
\]

\[
-1(n) = ( \quad ) n^2 + ( \quad ) n + ( \quad )
\]
<table>
<thead>
<tr>
<th>Summary</th>
<th>$T(n)$</th>
<th>Asymptotic Rate of Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>$o(n)$</td>
<td>$\Theta(n)$</td>
</tr>
<tr>
<td>Worst</td>
<td>$cn^2 + n$</td>
<td>$\Theta(n^2)$</td>
</tr>
<tr>
<td>Avg</td>
<td>$fn^2 + gn + h$</td>
<td>$\Theta(n^2)$</td>
</tr>
</tbody>
</table>

($a - h$ depend on $c_1 \ldots c_7$)
Illustrate asym. growth rate

A, B, C, D are algorithms that solve same problem (all worst-case):

<table>
<thead>
<tr>
<th>runtime</th>
<th>asymp. run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n^2$</td>
<td>$\Theta(n^2)$</td>
</tr>
<tr>
<td>$10n^2$</td>
<td>$\Theta(n^2)$</td>
</tr>
<tr>
<td>$10n^2+2n+10$</td>
<td>$\Theta(n^2)$</td>
</tr>
<tr>
<td>$1000n+10000$</td>
<td>$\Theta(n)$</td>
</tr>
</tbody>
</table>

Note: D is inferior for small n, but better for large sizes. Every line is eventually below every parabola.
Also lower order terms in $C$ are negligible for large $n$

$$\frac{C}{B} = (1 + \frac{1}{5n} + \frac{1}{n^2}) \rightarrow 1$$

Also we don't distinguish $A$ from $B$ since they can be equalized by running $B$ on a faster machine.
Informal defn of $\Theta(g(n))$
Drop lower order terms in $g(n)$, and replace coeff at highest order term by 1.

MergeSort: Divide and Conquer

Problem instance $I$

Sub-instances $I_1, I_2, \ldots, I_k$

Sub-instance solutions $S_1, S_2, \ldots, S_k$

Solution to $I$
A[p \ldots \ r] \quad \text{unsrted}

\begin{center}
\begin{tikzpicture}
  \node (root) at (0,0) {$A[p \ldots \ r]$};
  \node (left) at (-3,-3) {$A[p \ldots \ t]$};
  \node (right) at (3,-3) {$A[q+1 \ldots \ r]$};
  \node (merge) at (0,-6) {$A[p \ldots \ r]$};

  \draw[->] (root) -- (left) node[midway, above] {\text{sort}};
  \draw[->] (root) -- (right) node[midway, above] {\text{sort}};
  \draw[->] (left) -- (merge) node[midway, above] {\text{merge}};
  \draw[->] (right) -- (merge) node[midway, above] {\text{merge}};

  \node at (5,0) {\text{divide}};
\end{tikzpicture}
\end{center}

Assume we have \text{merge}(A, p, q, r)
MergeSort(A, p, r)

1.) if p < r

2.) q = \left\lfloor \frac{p + r}{2} \right\rfloor

3.) MergeSort(A, p, q)

T(\frac{n}{2})

4.) MergeSort(A, q+1, r)

T(\frac{n}{2})

5.) Merge(A, p, q, r)

n-1

How is Merge() accomplished?

\begin{align*}
\text{sorted} & \quad & \text{sorted} \\
A[p \ldots q] & \quad & A[q+1 \ldots r] \\
\downarrow \text{copy} & \quad & \downarrow \text{copy} \\
\text{temp} & \quad & \text{temp} \\
T[p \ldots q] & \quad & T[q+1 \ldots r] \\
\downarrow & \quad & \uparrow \text{merge} \\
A[p \ldots r] & \quad & \end{align*}
\[ n \text{ worst case # of comparisons by \texttt{mergeSort(\(A\)) \ in}} \]
\[ \text{length}(A[p..r]) - 1 \]
\[ = (r - p + 1) - 1 = r - p \]

\[ n \text{ worst case # of comparisons by \texttt{mergeSort(\(A\)) on arrays of length } n \].} \]

\[ \text{Exercise: show that if } p=1, r=n \]
\[ \text{and } q = \left\lfloor \frac{1+n}{2} \right\rfloor, \text{ then} \]
\[ A[p..q-1] \text{ has len. } \left\lfloor \frac{n}{2} \right\rfloor \]
\[ A[q+1..r] \text{ has len. } \left\lfloor \frac{n}{2} \right\rfloor \]
\( T(n) = \begin{cases} 0 & n = 1 \\ T\left(\lfloor \frac{n}{2} \rfloor\right) + T\left(\lceil \frac{n}{2} \rceil\right) + (n-1) & n \geq 2 \end{cases} \)

**Simplifying assumption:** \( n \) is an exact power of 2: \( n = 2^k \).

We get:

\( T(n) = \begin{cases} 0 & n = 1 \\ 2T\left(\frac{n}{2}\right) + (n-1) & n \geq 2 \end{cases} \)

**Solution:** \( T(n) = n \log(n) - n + 1 \)