January 22, 2003

Sanjay Ranka

ANALYSIS OF ALGORITHMS
Merge Sort

1. Divide the problem into a number of subproblems.
2. Conquer the problems by solving them recursively, or for small
   subproblems, solve in a straightforward manner.
3. Combine the solutions to the subproblems into the solution of the

Divide and Conquer Method

Divide and Conquer (Part I)
As an example, for inputs $-4, 10, 12, -5, -7, 8, 3$, the answer is 22.

(positive)

For convenience, the maximum subsequence sum is 0 if all the integers are negative.

Given possibly negative integers $a[1], a[2], \ldots, a[n]$, find the maximum subsequence sum.
Maximum Subsequence Problem: Algorithm 1

Step 1. Set $MAXSUM = 0$.

Step 2. Calculate $s(i, j) = \sum_{k=i}^{j} a[k]$, and update the $MAXSUM$ by $s(i, j)$ if $s(i, j) > MAXSUM$; for $1 \leq i \leq j \leq n$.

Step 3. $MAXSUM$ is the answer.
Note that $s$ is:

$\begin{align*}
\mathcal{L} \varphi + (1 - \mathcal{L}, \iota) s &= [\gamma] \varphi \frac{\dot{z} = \ell}{z \xi} = (\mathcal{L}, \iota) s \\
(\varepsilon \mu) O &= \frac{9}{u \xi + \varepsilon u \xi + \xi u} = \\
\left\{ \varepsilon \iota + \varphi (\xi + u \xi) - (\xi + u \xi + \varepsilon u) \right\} \frac{\dot{z} = \ell}{z \xi} &= \\
\frac{\dot{z}}{(\xi + \iota - u)(1 + \iota + u)} \frac{\dot{z} = \ell}{z \xi} &= \\
(1 + \iota - \mathcal{L}) \frac{\dot{z} = \ell}{z \xi} &= 1 \frac{\dot{z} = \ell}{z \xi} \\
\text{Analysis: In } s(\iota, \iota') (\iota + \iota - \mathcal{L}) \text{ terms, therefore the complexity}
\end{align*}$
Maximum Subsequence Problem: Algorithm 2

Divide and Conquer (Part I)

Step 1. Divide the sequence into two equal halves.

Step 2. Find the $\text{MAX}_M$ of the left half; via this recursive method.

Step 3. Find the $\text{MAX}_M$ of the right half; via this recursive method.

Step 4. Find $\text{MAX}_M = \max \{ \text{MAX}_M(1), \text{MAX}_M(2) \}$ of the sequence.

We should be able to find the $\text{MAX}_M$ in either the left half or the right half of the sequence. For example, in -4, 12, -5, -7, 8, 3, the right half of the sequence is 12, -5, -7, 8, 3, and the left half is -4. We can recursively find the $\text{MAX}_M$ in both halves and take the maximum of the two to get the $\text{MAX}_M$ of the whole sequence.
is \( \sum_{1}^{7} \) due to sum of terms 1 through 7.

Here \( \sum_{1}^{7} \) is due to sum of first three terms,

\[ 4, -3, 5, -2, -1, 2, 6, 2. \]

What happens if the maximum sum is due to terms in the middle?
Step 1. Divide the sequence into two equal halves. Set $m = \lceil (1 + u)/2 \rceil$.

Step 2. Find the $\text{MAX SWM}(T)$ of the left half; via this recursive method.

Step 3. Find the $\text{MAX SWM}(R)$ of the right half; via this recursive method.

Step 4. If $\text{MAX SWM}(T) > \text{MAX SWM}(T)$, then

$$\text{MAX SWM}(T) = \text{MAX SWM}(T) .$$

Otherwise, if $\text{MAX SWM}(T) < \text{MAX SWM}(T)$, then

$$\text{MAX SWM}(T) = \text{MAX SWM}(T) .$$

Calculate $[? + i = \text{MAX SWM}(T) = \text{MAX SWM}(T)$.

For $i = 0$, set $m = 0$. Then

Method.

Maximum Subsequence Problem: Algorithm 3

Divide and Conquer (Part I)
\[ u + u \leq (u) L = (1) \] gives \( L = 1 \) 

With addition of the edges from the border to the centers, solution of this equation, of adding sums from the stacks, assignments etc. The term \( u \) represents the cost of managing of the stacks, assignments etc. The term \( u \) represents the cost of addition, then \( u + (z/u) \) \( \leq = (u) L \).

Here we have ignored the costs of addition, then \( u + (z/u) \) \( \leq = (u) L \).

Analysis: \( (u) L \) denotes the total number of basic operations of

\[ \left( (H)^{g \cap S} \times M + (T)^{g \cap S} \times M \right)^{\text{max}} = M \cap S \times M \]

Step 6. Finally, \( (H)^{g \cap S} \times M + (T)^{g \cap S} \times M \)

Step 4. As step 4 except for the right half of the sequence.

Divide and Conquer (Part I)
use the exact methods.  

If we want to find the exact solution of a recurrence equation we must

\[(I) \mathcal{L} u^p + \left(1 - \frac{w^2}{u}\right) f_{-1} u^p + \cdots + \frac{z^p}{u} f_0 + \frac{q}{u} f_0 + (u) f = (u) \mathcal{L}\]

Hence the total effort and solutions.

Here we will consider such equations in detail to find their

\[\text{for } \begin{cases} q < 0, \quad \text{for } 1' \end{cases} \quad (u f + \frac{q}{u} \mathcal{L} p = (u) \mathcal{L}\]

Divide and Conquer methods generally lead to the following recurrence

---

Divide and Conquer (Part I)
2. There are no reasons to be overly concerned with small variations.

1. There are no reasons to be overly concerned with the boundary conditions. All boundary conditions give results of the same order.

If we want asymptotic results then we don't want to worry about minor details, such as

- 

\[ \left( \frac{q}{u} \right) f \]
- 

\[ \left( \frac{q}{u} \right) f \]
- 

\[ \left( \frac{q}{u} \right) f \]

\[ \left( \frac{q}{u} \right) f \]
solution approximates the desired solution.

Extending the definition of \( f \) to real numbers and hope that this

even integer. General idea is to first solve the equation by

an equals to algorithms when \( n \), the size of the problem is not an

approximated by \( \lfloor \frac{\sqrt{n}}{\log n} \rfloor \), even though later does not make sense in

of the recurrence equation. For example, can be

Divide and Conquer (Part 1)
\[(u)f \Theta = (u)\mathcal{L}\ 	ext{for some constant } c \geq 1 \text{ for large } u, \text{ then}
\]
\[(u)f \geq (q/u)f \mathcal{L} \text{ and if } (\log q \log u)^2 = (u)f \mathcal{L} \]
\[(u \log q \log u) \Theta = (u)\mathcal{L} \text{ then } (u \log q \log u) \Theta = (u)f \mathcal{L} \]
\[(u \log q \log u) \Theta = (u)\mathcal{L} \text{ otherwise for some constant } (c-1 \log q \log u) O = (u)f \mathcal{L} \]

where \( \frac{q}{u} \) is to be interpreted as either \( \left\lfloor \frac{q}{u} \right\rfloor \) or \( \left\lceil \frac{q}{u} \right\rceil \).

\[\left( (u)f + \left( \frac{q}{u} \right)\mathcal{L} \right) \mathcal{L} = (u)\mathcal{L} \]

\[\text{The Master Theorem}\]

\[\text{Divide and Conquer (Part I)}\]
\[
\begin{align*}
\left( \frac{1}{\varrho} \right)^{\varpi} \left( 1 - \varrho \right)^{\varpi} O = \\
\left( \frac{1}{\varrho} \right)^{\varpi} \left( 1 - \varrho \right)^{\varpi} O = \\
\left( \frac{1}{\varrho} \right)^{\varpi} \left( 1 - \varrho \right)^{\varpi} O = (u) b
\end{align*}
\]

I. Let \( \in (0, \infty) \). \( \varrho \) is an integer.

We consider in greater detail.

\[
\left( \frac{1}{u} \right)^{\varpi} \left( 1 - \varrho \right)^{\varpi} O = (u) f
\]

I. Let \( \in (0, \infty) \). \( \varrho \) is an integer.

**Proof of the Master Theorem**

Divide and Conquer (Part I)
\[ u(n_{q_{01}}u) \circ = \]
\[ \frac{v^{\mathcal{Q}}_{q_{01}} u_{\mathcal{P}}}{v^{\mathcal{Q}}_{q_{01}} u_{\mathcal{P}}} \circ = \]
\[ \left( \frac{v^{\mathcal{Q}}_{q_{01}} u_{\mathcal{P}}}{v^{\mathcal{Q}}_{q_{01}} u_{\mathcal{P}}} \right)_{1-w}^{0}\circ \geq (u)\delta \]

Consider for an upper bound.

\[ (n_{q_{01}}u) \circ \geq (u)f \geq v^{\mathcal{Q}}_{q_{01}} u_{\mathcal{P}} \circ \]

2. Let \( \Theta = (u)f \) and

\[ (v^{\mathcal{Q}}_{q_{01}} u)O = (u)O \Theta = (u)f \Theta = (u)O \]

\[ ((u)O)_{\mathcal{P}} = (u)O_{\mathcal{P}}^{\mathcal{Q}_{01}} \]

\[ (1-q)_{\mathcal{P}} = (1-q)_{\mathcal{P}}^{\mathcal{Q}_{01}} \]

Divide and Conquer (Part I)
\[(u)f \geq \left( \frac{q}{u} \right) f\]

In general,

\[\left( \frac{q}{u} \right) f \geq \left[ \left( \frac{q}{u} \right) f v \right] = \left[ \left( \frac{q}{u} \right) f \right] v \geq \left( \frac{q}{u} \right) f v = \left( \frac{q}{u} \right) f v\]

Then

\[q < u, \text{ and a fixed } c, \text{ \ for all values of } u \text{ for } (u)f \geq \left( \frac{q}{u} \right) f v\]

3. Suppose that satisfies the condition

\[u^{q}(u^{q})^{\Theta} = (u)b\]

Similar result for the other case gives

Divide and Conquer (Part I)
Lemma: Let $a < q$ and let $I$ be constants, and let $f$ be a $(u)f$. Hence, in summary we have

$$(u)f \log_{q+\epsilon} I = (u)f \quad \text{implies that} \quad (u)f \geq \left( \frac{q}{u} \right)f/a$$

Thus, $((u)f)_{\Theta} = (u)b$, i.e., $(u)f_{\Theta}^{2-1} \geq (u)b \geq (u)f$. It can be shown that

$$\text{constant} = \frac{c - I}{I} \geq \frac{c - I}{u^2 - I}$$

Now suppose that $c > I$, then

$$(u)f \frac{I - c}{I - u^2} =$$

$$(u)f \frac{0}{I - u} \geq$$

$$(\frac{uq}{u})f_{1-w} + \cdots + (\frac{q}{u})f + (u)f =$$

$$f \geq (u)f$$

So,
\[((u)f)\Theta = (u)_L\]

for some constant \(c > 1\) and sufficiently large \(u\), then

\[((u)f)c \geq (q/u)f^q\]

and if

\[3. \text{ If for some constant } c \neq 0', \text{ and if} \]

\[(u \bigtriangledown_{q_{0}^{\text{q}_{0}}} u) \Theta = (u)_L \text{ then} \]

\[(v_{q_{0}^{\text{q}_{0}}} u) \Theta = (u)_L \]

\[2. \text{ If for some constant } c \neq 0' \text{ then} \]

\[(v_{q_{0}^{\text{q}_{0}}} u) O = (u)f \text{ if as follows:} \]

asymptotically for exact powers of \(q\) of a positive integer. Then can be bounded

\[\text{where } q \text{ is a positive integer. Then } (u)_L \text{ can be bounded} \]

\[\begin{align*}
\dot{q} &= u \not\vdash (u)f + \left(\frac{q}{u}\right)_L v \\
\vdash 1 &= u \not\vdash (1)v \not\vdash (1)\Theta \end{align*}\]

= \((u)_L\)

exact powers of \(q\) by the recurrence

\[\text{nonnegative function defined on exact powers of } u. \text{ Define } (u)_L \text{ on} \]

Divide and conquer (Part I)
Example Application: Merge Sort

1. Divide in two parts as equal as possible.

2. Merge?
Continue in this manner.

In case \( (!!) \) compare \([1]X\) and \([2]Y\)

In case \( (!) \) compare \([1]X\) and \([2]Y\)


Compare \([1]X\) and \([1]Y\)

Procedure for Merging

\[
\begin{cases}
\end{cases}
\]

Merge Given:

---

Merging Algorithm

Divide and Conquer (Part I)
two sorted sequences of order \( n \).

\[
(u) \Theta + (\lfloor \frac{2}{n} \rfloor) L + (\lceil \frac{2}{n} \rceil) L = (u) L
\]

If \( (u) L \) denotes the total time to sort, then

---

The unsorted sequence

\[
\begin{array}{cccccccc}
\text{6} & \text{3} & \text{2} & \text{1} & \text{6} & \text{4} & \text{2} & \text{3} \\
\text{2} & \text{6} & \text{1} & \text{3} & \text{4} & \text{6} & \text{3} & \text{2} \\
\text{1} & \text{2} & \text{3} & \text{6} & \text{1} & \text{2} & \text{3} & \text{4} \\
\end{array}
\]
Asymptotic analysis: Using Master theorem we have

\[
\begin{align*}
\left(\frac{\log n}{\log u}\right) \Theta &= \left(\frac{\log n}{\log u}\right) L \\
\left(\frac{\log n}{\log u}\right) L &= \left(\frac{\log n}{\log u}\right) L \\
\left(\frac{\log n}{\log u}\right) L &= \left(\frac{\log n}{\log u}\right) L
\end{align*}
\]

Worst Case: (In terms of only \# of comp.)

Analysis of Merge Sort

Dive and Conquer (Part I)