MASTER THEOREM

Thursday, September 27, 2018 5:20 PM

Generalizing the MERGE SORT recurrence:

$$C(1) = c_0$$
$$C(n) = aC\left(\frac{n}{b}\right) + f_n, \text{ for } n = b^m$$

[In the case of merge sort, $c_0 = 0$, $a = 2$, $b = 2$, $f_n = n-1 \in \Theta(n^1)$]

$$C(n) = C(b^m) = aC\left(\frac{b^m}{b}\right) + f_{b^m} = aC(b^{m-1}) + f_{b^m}$$

Rewriting the recurrence in terms of m

$$C(1) = c_0$$
$$C(b^m) = aC(b^{m-1}) + g_m \quad (g_m = f_{b^m})$$

"Unrolling" the recurrence

$$\begin{aligned}
C(b^m) &= aC(b^{m-1}) + g_m \\
&= a\left[aC(b^{m-2}) + g_{m-1}\right] + g_m = a^2C(b^{m-2}) + a^1 g_{m-1} + a^0 g_m \\
&= a^2\left[aC(b^{m-3}) + g_{m-2}\right] + a^1 g_{m-1} + a^0 g_m \\
&= a^3 C(b^{m-3}) + a^2 g_{m-2} + a^1 g_{m-1} + a^0 g_m \\
&= \cdots\cdots \\
&= \underbrace{a^m C(b^0=1)}\underbrace{+ a^{m-1} g_1 + a^{m-2} g_2 + \cdots\cdots + a^1 g_{m-1} + a^0 g_m} \\
&= a^m c_0 + \sum_{i=0}^{m-1} a^i g_{m-i}
\end{aligned}$$

Assume that $g_m \in \Theta\left[(b^m)^d\right]$ for some $d \geq 0$

$g_m = f_{b^m} = f_n \in \Theta(n^d)$ for some $d \geq 0$.

MASTER THEOREM: Suppose $C(1) = c_0$

$$C(n) = aC\left(\frac{n}{b}\right) + f_n, \; n = b^m, \text{ when } f(n) \in \Theta(n^d), d \geq 0$$

$a \geq 1$, $b \geq 2$, $c_0 \geq 0$, $d \geq 0$

There are 3 cases

1) $a = b^d$, $C(n) \in \Theta(n^d \lg n)$
2) $a < b^d$, $C(n) \in \Theta(n^d)$
3) $a > b^d$, $C(n) \in \Theta(n^{\log_b a})$

APPLYING TO MS: $C(1) = 0$

$C(n) = 2C\left(\frac{n}{2}\right) + n-1$, $n = 2^m$ $\quad c_0 = 0, a = 2, b = 2, d = 1$

Applying the MT, $a = 2 = 2^1 = b^d$, $C(n) \in \Theta(n^1 \lg n)$

Proof: Recall that

$$\boxed{C(b^m) = c_0 a^m + \sum_{i=0}^{m-1} a^i f_{b^{m-i}}}$$

$$(b^{m-i})^d = \frac{(b^m)^d}{(b^i)^d}$$

When $f_n = \Theta(n^d)$

$$C(b^m) = c_0 a^m + \sum_{i=0}^{m-1} a^i (b^{m-i})^d = c_0 a_m + (b^m)^d \sum_{i=0}^{m-1} \frac{a^i}{(b^d)^i}$$

$$C(b^m) = c_0 a^m + \sum_{i=0} a^i \left(b^{m-i}\right)^d = c_0 a_m + (b^m)^d \sum_{i=0} \frac{a^i}{(b^d)^i}$$

$$= c_0 a^m + (b^d)^m \sum_{i=0}^{m-1} \left(\frac{a}{b^d}\right)^i$$

$$= c_0 a^m + (b^d)^m \left[\frac{1 - \left(\frac{a}{b^d}\right)^m}{1 - \frac{a}{b^d}}\right], \quad \frac{a}{b^d} \neq 1$$

$$= c_0 a^m + \frac{(b^d)^m - a^m}{1 - \frac{a}{b^d}}$$

Three cases:

1) $a = b^d$: $C(b^m) = c_0 a^m + m(b^m)^d = c_0 (b^m)^d + m(b^m)^d$

$C(n) = c_0 n^d + \lg n \; n^d \in \Theta(n^d \lg n)$

2) $a < b^d$: $\underbrace{c_0 a^m + \left(\frac{1}{1-\frac{a}{b^d}}\right)\left[(b^d)^m - a^m\right]}_{< \Theta((b^d)^m) = \Theta(n^d)}$ $\quad a < b^d$, $a^m < (b^d)^m$

3) $a > b^d$

$$c_0 a^m + \frac{1}{(1-\frac{a}{b^d})}\left((b^d)^m - a^m\right) \in \Theta(a^m)$$

$$= \Theta\left(\left(b^{\log_b a}\right)^m\right)$$

$$= \Theta\left((b^m)^{\log_b a}\right)$$

$$= \Theta\left(n^{\log_b a}\right)$$

NOTE: The proof presented DOES NOT work when $f \notin \Theta(n^d)$ for some $d \geq 0$.

For example, $f_n = n \lg n$

APPLICATION 1: Binary Search

```
BS ( A[lo..hi], x)      // A[lo..hi] is sorted
{
    if (lo > hi)  // n = 0
        return false;
    m = (lo+hi)/2;   // lo + (hi-lo)/2 to avoid overflow
    switch ( comp (A[m], x) )
    {
        case 0:  return true;
        case +:  return BS (A[lo..mid-1], x);
        case -:  return BS (A[mid+1..hi], x);
    }
```

}

worst-case
$C(n) =$ # comp( ) performed by BS on arrays of size n

$C(1) =$ const
$C(n) = 1 + C(\frac{n}{2})$, $n = 2^m$

$a = 1$, $b = 2$, $d = 0$ because $1 \in \Theta(n^0)$
1 vs 2^0 $1 = 2^0$ so $C(n) \in \Theta(n^0 \lg n) = \Theta(\lg n)$

ADDITION:
Input: 2 n-bit numbers a, b
Output: a+b

Ex:
$$\begin{array}{r} {}^{1\,1} \\ 1234 \\ +\,5678 \\ \hline 6912 \end{array}$$
$\Theta(n)$ digit operations

MULTIPLICATION
Input: 2 n-bit numbers a, b
Output: a*b

$$\begin{array}{r} 1234 \\ *\,5678 \\ \hline \end{array}$$
$\Theta(n)$
$\Theta(n)$
⋮
$\Theta(n)$
$\Theta(n^2)$

MERGE SORT IDEA
12 | 34
56 | 78
$n = 4$

$1234 \times 5678 = (12 \times 10^2 + 34) \cdot (56 \times 10^2 + 78)$
$= \boxed{12 \times 56 \times 10^4} + (12 \times 78 \times 10^2 + 34.56 \times 10^2) + \boxed{34.78}$

$(12 \times 78 + 34 \times 56) \times 10^2$

$C(n) = 4C(\frac{n}{2}) + \Theta(n)$

$a = 4$, $b = 2$, $d = 1$
a v b^d
4 v $2^1 = 2$
$C(n) \in \Theta(n^{\log_2 4} = n^2)$

If we can reduce # of recursive calls (a), we can get a subquadratic alg!!!

<u>Idea</u>:

1) 12×56 (1 recursive call)
2) 34×78 (1 recursive call)
3) $(12+34) * (56+78)$ (1 recursive call)

$$(12+34) \cdot (56+78) = \cancel{12 \cdot 56} + 12 \cdot 78 + 34 \cdot 56 + \cancel{34 \cdot 78}$$
$$- \cancel{12 \cdot 56} - \cancel{34 \cdot 78}$$

$$c(n) = 3c\left(\tfrac{n}{2}\right) + \Theta(n)$$

$a = 3,\ b = 2,\ d = 1 \qquad 3 > 2^1, \quad c(n) \in \Theta\left(n^{\log_2 3 = 1.5\ldots}\right)$