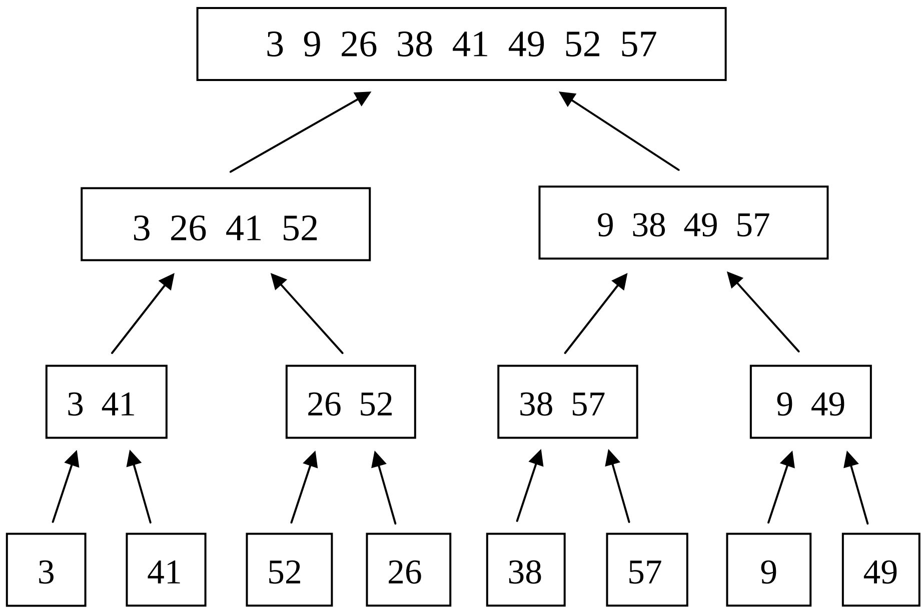
Solutions to CS / MCS 401 Exercise Set #3 (Fall 2007)

Exercise 2.3-1



Exercise D.

C(0) = 0 = $d \cdot 0$, so the result holds when $n = 0$.

Let $n \geq 1$, and assume the result holds for all i with $i < n$. $C(n) = d + C(k) + C(n-k)$ where $0 \leq k \leq n-1$. Note $n-k-1 \leq n-1$. By the inductive hypothesis,

$$C(n) = d + dk + d(n-k-1) = d(1 + k + (n-k-1)) = dn,$$

so the result also holds for n. By induction it holds for all nonnegative integers.

Exercise E.

Let L, M, and R be sorted arrays of length $n/3$ (possibly $\lfloor n/3 \rfloor$ or $\lceil n/3 \rceil$, so the sum of the lengths is n). For simplicity, assume n is a power of 3, so always L, M, and R have length $n/3$. Assume that each array has an extra element ∞ at the end. We can merge L, M, and R into a single sorted array A of length n using the algorithm below. Here i, j, and k represent the positions of the current elements in L, M, and R respectively; and x represents the smallest element not yet merged from M or R, provided *xValid* is true. As usual, indentation indicates nesting of blocks.

```
i = 1;  j = 1;  k = 1;
xValid = false;
for ( q = 1,2, ..., n )
    if ( not xValid )
        if ( M[j] ≤ R[k] )            (*)
            x = M[j];
            j = j + 1;
        else
            x = R[k];
            k = k + 1;
        xValid = true;
    if ( L[i] ≤ x )                   (**)
        A[q] = L[i];
        i = i + 1;
    else
        A[q] = x;
        xValid = false;
```

Comparisons are performed in the lines (*) and (**). The comparison in line (**) is performed on each pass through the loop — a total of n times. The comparison on line (*) is always performed on the first pass ($q = 1$). On the remaining passes, it is performed if the element merged to A on the previous pass came from M or R, but *not* if it came from L. Thus the total number of comparisons in line (*) is

$$n - (\text{number of elements merged from L on the first } n{-}1 \text{ passes})$$
$$= n - (n/3 \text{ or } n/3-1)$$
$$= 2/3\,n \;\text{ or }\; 2/3\,n + 1$$

times. The total number of comparisons performed by the algorithm is $5/3\,n$ or $5/3\,n + 1$.

Exercise F.

$C(n) = 3C(n/3) + 5/3n$, $C(1) = 0$. We assume $n = 3^k$, so $k = \log_3(n)$.

This recurrence has the correct form for the Master Theorem with $a = 3$, $b = 3$, $E = 1$, $n^E = n$, $f(n) = 5/3n$. However, the Master Theorem tells us only that the solution is $\Theta(n\log_3(n))$, whereas we are asked for an exact solution. In the proof of the Master Theorem, we showed that

$$C(n) = f(n) + af(n/b) + a^2 f(n/b^2) + ... + a^{k-1} f(n/b^{k-1}) + a^k d, \text{ where } d = C(1).$$

Substituting the appropriate values for a, b, $f(n)$, and d, we obtain

$$C(n) = 5/3n + 3(5/3)(n/3) + 3^2(5/3)(n/3^2) + ... + 3^{k-1}(5/3)(n/3^{k-1}).$$

This sum contains k terms, each equal to $5/3n$, so the sum is $5/3nk = 5/3n\log_3(n)$.

The exact solution when n is a power of 3 is $\mathbf{C(n) = 5/3\,n\log_3(n) \approx 1.052\,n\lg(n)}$.

By contrast, ordinary (2-way) mergesort uses approximately $n\lg(n)$ comparisons.