S. Schwoon / D. Suwimonteerabuth | Winter Term 2008/09

Fundamental Algorithms

Solution Keys 5

1. We consider the case where n is an even number. (We only need to split a and b into $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$ digits if n is an odd number.) The multiplication becomes

$$ab = 10^n ux + 10^{n/2}(uy + vx) + vy \ .$$

At first glance, it seems that we reduce a single multiplication of size n into *four* multiplications of size $n/2$. However, with the following observation the term $(uy + vx)$ can be computed by a single multiplication, given that the values of ux and vy are known:

$$uy + vx = (u + v)(x + y) - ux - vy \ .$$

Therefore, we can compute a multiplication of size n by computing *three* multiplications of size $n/2$. Let $t(n)$ be the time required for multiplying two n-digit numbers. We have

$$t(n) = 3t(n/2) + g(n) \ ,$$

where $g(n) \in \Theta(n)$ is the time required for shifts and additions. It can be shown that $t(n) \in \Theta(n^{\log 3})$, which is better than the time complexity of long multiplication.

2. Since each comparison can exclude at most half of relative orderings of elements and there can be $\frac{n!}{2^n}$ possible inputs, any comparison sort needs to make at least

$$\log_2(n!) - n = \Theta(n \log n) - n = \Theta(n \log n)$$

comparisons.

3. The following procedure `preprocess` uses the idea of the counting sort algorithm. The first two loops count the number of occurrences of $a[i]$ for each i between 1 and n. The last loop accumulates the results: $b[i]$ therefore memorizes the number of elements between 1 and i.

Procedure preprocess
Input: array a of length n, whose elements are in the range $[1, k]$
Output: array b

for $i = 1$ **to** k **do** $b[i] := 0$ **od**;
for $i = 1$ **to** n **do** $b[a[i]] := b[a[i]] + 1$ **od**;
for $i = 2$ **to** k **do** $b[i] := b[i] + b[i-1]$ **od**;
return b;

With array b, the procedure `query` finds the number of elements between u and v in a constant time.

Procedure query
Input: array b, integers u and v in the range $[1, k]$
Output: the number of elements in a between u and v

if $u = 1$ **then return** $b[v]$ **fi**;
return $b[v] - b[u-1]$;