BLACKBOARD PROOFS

CSE202 – WEEK 1

1. Cubic complexity for matrix multiplication

If A and B are two $n \times n$ matrices and $C = AB$, then the entry (i, j) of C is

$$\sum_{k=1}^{n} a_{i,k} b_{k,j}.$$

Using this formula for the computation requires n multiplications and $n-1$ additions. Doing it for each of the n^2 entries of C brings the complexity to $O(n^3)$ operations (multiplications, additions) on the coefficients.

2. The number of multiplications of binary powering

The aim is to prove that if $C(n)$ is the sequence defined by

$$C(n) = 1 + \begin{cases} C(n/2), & \text{for even } n > 0, \\ C((n-1)/2) + 1, & \text{for odd } n > 1 \end{cases} \qquad \text{with } C(0) = C(1) = 0,$$

then the following lemma holds.

Lemma 1. *For $n \geq 1$,*

$$C(n) = \lfloor \log_2 n \rfloor - 1 + \lambda(n)$$

where $\lambda(n)$ is the number of 1's in the binary expansion of n.

First, it might be useful to recall a few properties of binary (or base 2) expansions. They are very similar to base 10, only less familiar. If $n \geq 1$, its binary expansion always starts with a 1. The last bit is 0 if n is even and 1 otherwise. As in base 10, multiplying by the base (here 2) amounts to adding a 0 at the end. For instance, for $k \geq 0$, the expansion of 2^k is a 1 followed by k 0's. For such a number, we have $\log_2 2^k = k$. Thus since the logarithm is increasing, for any n such that $2^k \leq n < 2^{k+1}$, taking the logarithm gives $k \leq \log_2 n < k+1$ and thefore $\lfloor \log_2 n \rfloor = k$, which is the length of the binary expansion, minus 1.

Also, $\lambda(2m) = \lambda(m)$ since multiplying by 2 only adds a 0 at the end; then adding an extra 1 gives $\lambda(2m+1) = \lambda(m) + 1$.

Proof of the Lemma. The proof is by induction.

First, $n = 1$ has $\overline{1}^2$ for its binary expansion, thus $\log_2 1 = 0$ and $\lambda(1) = 1$ so that both sides of the equality agree.

Next, assume the property holds for $k = 1, \ldots, n-1$. Let $m = n/2$ if n is even and $m = (n-1)/2$ otherwise. Thus $1 \leq m \leq n-1$ and the property holds for m.

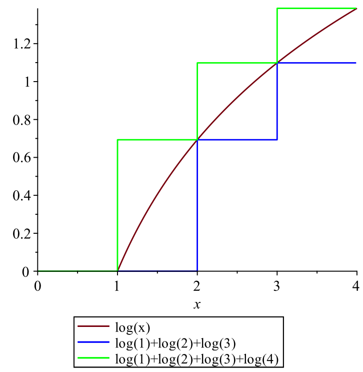

FIGURE 1. Comparison of sum and integral

If $k = \lfloor \log_2 m \rfloor$, then $2^k \leq m \leq 2^{k+1} - 1$ implies $2^{k+1} \leq 2m \leq 2m + 1 < 2^{k+2}$ so that $\lfloor \log_2 n \rfloor = k + 1$. Since $1 \leq m \leq n - 1$, the induction hypothesis holds, so that

$$
\begin{aligned}
C(m) &= \lfloor \log_2 m \rfloor - 1 + \lambda(m) \\
&= \begin{cases} \lfloor \log_2(2m) \rfloor - 2 + \lambda(2m) = C(n) - 1 & \text{if } n \text{ is even,} \\ \lfloor \log_2(2m+1) \rfloor - 2 + \lambda(2m+1) - 1 = C(n) - 2 & \text{otherwise,} \end{cases}
\end{aligned}
$$

and thus the induction is proved. □

Another possible expression of the proof is to define $D(n)$ as the right-hand side in the Lemma and then show, by the same reasoning, that it satisfies the same recurrence as $C(n)$, with the same initial condition and finally conclude by observing that this recurrence has a unique solution with a given initial condition.

3. Minimal number of comparisons to sort n elements

The number of comparisons must be sufficient to tell apart each of the $n!$ distinct possible permutations of n distinct elements. The optimal algorithm can be seen as navigating in a tree, whose root is the set of $n!$ permutations and at each stage, a comparison splits into two groups the remaining permutations. This makes a binary tree with $n!$ leaves. The number of comparisons performed by the algorithm is the height of the tree. This cannot be smaller than the height of a perfectly balanced tree with $n!$ leaves, which is $\lfloor \log_2 n! \rfloor$ (by induction).

To obtain the asymptotic behaviour, observe that by comparing sum and integral (see Figure 1), it follows that

$$\int_1^n \log x \, dx \leq \log(n!) = \log 1 + \cdots + \log n \leq \int_1^{n+1} \log x \, dx.$$

Since a primitive of $\log x$ is $x \log x - x$, it follows that both sides are asymptotically equivalent to $n \log n$ as $n \to \infty$. Multiplying by $1/\log 2$ gives that $n \log_2 n$ is therefore an asymptotic lower bound on the number of required comparisons.