6 Recurrences

Algorithm 2 mergesort(list L)

```
s ← size(L)
if s ≤ 1 return L
L1 ← L[1 ··· ⌊s/2⌋]
L2 ← L[⌈s/2⌉ ··· n]
mergesort(L1)
mergesort(L2)
L ← merge(L1, L2)
return L
```

This algorithm requires

$$T(n) \le 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + \mathcal{O}(n)$$

comparisons when $n > 1$ and 0 comparisons when $n \le 1$.

Recurrences

How do we bring the expression for the number of comparisons ($\approx$ running time) into a closed form?

For this we need to solve the recurrence.

Methods for Solving Recurrences

1. **Guessing+Induction**
   Guess the right solution and prove that it is correct via induction. It needs experience to make the right guess.
2. **Master Theorem**
   For a lot of recurrences that appear in the analysis of algorithms this theorem can be used to obtain tight asymptotic bounds. It does not provide exact solutions.
3. **Characteristic Polynomial**
   Linear homogenous recurrences can be solved via this method.

6.1 Guessing+Induction

First we need to get rid of the $\mathcal{O}$-notation in our recurrence:

$$T(n) \le \begin{cases} 2T(\lceil \frac{n}{2} \rceil) + cn & n \ge 2 \\ 0 & \text{otherwise} \end{cases}$$

Assume that instead we had

$$T(n) \le \begin{cases} 2T(\frac{n}{2}) + cn & n \ge 2 \\ 0 & \text{otherwise} \end{cases}$$

One way of solving such a recurrence is to guess a solution, and check that it is correct by plugging it in.

6.1 Guessing+Induction

Suppose we guess $T(n) \le dn \log n$ for a constant d. Then

$$\begin{aligned} T(n) &\le 2T\left(\frac{n}{2}\right) + cn \\ &\le 2\left(\frac{n}{2}\log\frac{n}{2}\right) + cn \\ &= dn(\log n - 1) + cn \\ &= dn\log n + (c-d)n \\ &= dn\log n \end{aligned}$$

if we choose $d \ge c$.

Formally one would make an induction proof, where the above is the induction step. The base case is usually trivial.

6.1 Guessing+Induction

$$T(n) \le \begin{cases} 2T(\frac{n}{2}) + cn & n \ge 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \le dn \log n$.
Proof. (by induction)

- **base case** $(2 \le n < 16)$: true if we choose $d \ge b$.
- **induction step** $2 \ldots n-1 \to n$:

  Suppose statem. is true for $n' \in \{2, \ldots, n-1\}$, and $n \ge 16$.
  We prove it for n:

$$\begin{aligned} T(n) &\le 2T\left(\frac{n}{2}\right) + cn \\ &\le 2\left(\frac{n}{2}\log\frac{n}{2}\right) + cn \\ &= dn(\log n - 1) + cn \\ &= dn\log n + (c-d)n \\ &= dn\log n \end{aligned}$$

- Note that this proves the statement for $n \in \mathbb{N}_{\ge 2}$, as the statement is wrong for $n = 1$.
- The base case is usually omitted, as it is the same for different recurrences.

Hence, statement is true if we choose $d \ge c$.

6.1 Guessing+Induction

Why did we change the recurrence by getting rid of the ceiling?

If we do not do this we instead consider the following recurrence:

$$T(n) \le \begin{cases} 2T(\lceil \frac{n}{2} \rceil) + cn & n \ge 16 \\ b & \text{otherwise} \end{cases}$$

Note that we can do this as for constant-sized inputs the running time is always some constant (b in the above case).

6.1 Guessing+Induction

We also make a guess of $T(n) \le dn \log n$ and get

$$\begin{aligned} T(n) &\le 2T\left(\left\lceil\frac{n}{2}\right\rceil\right) + cn \\ &\le 2\left(d\left\lceil\frac{n}{2}\right\rceil\log\left\lceil\frac{n}{2}\right\rceil\right) + cn \\ \boxed{\lceil \tfrac{n}{2} \rceil \le \tfrac{n}{2} + 1} \quad &\le 2\big(d(n/2+1)\log(n/2+1)\big) + cn \\ \boxed{\tfrac{n}{2} + 1 \le \tfrac{9}{16}n} \quad &\le dn\log\left(\frac{9}{16}n\right) + 2d\log n + cn \\ \boxed{\log \tfrac{9}{16}n = \log n + (\log 9 - 4)} \quad &= dn\log n + (\log 9 - 4)dn + 2d\log n + cn \\ \boxed{\log n \le \tfrac{n}{4}} \quad &= dn\log n + (\log 9 - 3.5)dn + cn \\ &\le dn\log n - 0.33dn + cn \\ &\le dn\log n \end{aligned}$$

for a suitable choice of d.