1. Do’s

2. Don’t’s
DO’S
WHAT DO WE EXPECT?

• You should be able to apply standard techniques to design and
  analyze simple algorithms.

• This means:
  • You describe an *algorithmic idea*.
  • You prove that that idea is *correct*.
  • You *analyze the complexity* of a specific implementation of that
    idea, possibly explaining further data structures needed to
    achieve it.

• Note: there’s *not necessarily a pseudo-code* (but it can help).

• Note: this lecture is *not about programming*, it’s about ideas.
DO'S
ANALYZING A RECURRENCE 1

- Explain what you will do, do it, conclude.
- You don’t need to be verbose.

\[
3(a) T(m) = 3T\left(\frac{m}{2}\right) + O(m)
\]

Wondo o teorema mestre:

\[
T(m) = aT\left(\frac{m}{b}\right) + O(m^k)
\]

\[
a = 3 \\
b = 2 \\
k = 1
\]

\[
T(m) = \begin{cases} 
  m^k \log m, & \text{se } a = b^k \\
  m^k, & \text{se } a < b^k \\
  m \log b^k, & \text{se } a > b^k 
\end{cases}
\]

\[
3 > 2 \quad (3^{rd\, case})
\]

\[
T(m) = \Theta(m \log^2 m)
\]
- You don't know how to solve it, but have a strong intuition: go with it, *give a proof by induction*.

\[
T(n) = T(n/2) + T(n/4) + O(n)
\]

Since the recurrence covers \(\frac{3}{4}\) of the size of the input, probably the total is dominated by \(O(n)\).

**Hyp.** \(T(n) = cn = O(n)\).

**Proof.** By induction on \(n\).

\[
T(n) \leq T(n/2) + T(n/4) + Cn
\]

\[
\leq cn/2 + cn/4 + Cn
\]

\[
\leq cn/2 + cn/4 + cn/4 \quad \text{for} \quad C \leq C/4
\]

\[
= cn.
\]
DO'S AND DONT'S

PRESENT A CONCLUSION WITHOUT ARGUMENTS

3) \[ T(m) = 3T\left(\frac{m}{2}\right) + O(m) \]

\[ a \quad b \quad f(m) \]

\[ m = O\left(m \log^2 \right) \]
\[ m = O(m^1) = 7 \Theta\left(m \frac{\log^3}{2}\right) \]

\[ T(m) = \Theta\left(m \frac{\log^3}{2}\right) \]
DONT'S

SKIP OVER ESSENTIAL STEPS WITH NO EXPLANATION

DO'S AND DONT'S

Question 03:

\[ T(n) = 3T\left(\frac{m}{2}\right) + O(n) \]

\[ \sum_{i=0}^{\log_3 n} 3 \cdot \left(\frac{m}{2}\right)^i = \sum_{i=0}^{\log_3 n} \left(\frac{1}{2}\right)^i + \sum_{i=0}^{\log_3 n} \left(\frac{1}{2}\right)^{2i} + \sum_{i=0}^{\log_3 n} \left(\frac{1}{2}\right)^{2i+1} = \sum_{i=0}^{\log_3 n} \left(\frac{1}{2}\right)^i \]

\[ = \sum_{i=0}^{\log_3 n} \left(\frac{1}{2}\right)^i = \log_3 n \]

[0(\log n)]
DO'S AND DONT'S

THE ENGINEERING APPROACH: JUST DO SOMETHING, IT MAY WORK

\[ T(n) = T(n/2) + T(n/4) + O(n) \]

\[ T(n) = \sum_{i=0}^{\log n} \frac{n}{2^i} + \sum_{i=0}^{\log n} \frac{n}{4^i} = O(n) + O(n) \]

\[ T(n) = O(n^2) \]
DONT'S
PRESENT AN ALGORITHM WITHOUT COMMENT

2) \(Palim\ (C, m)\)

For \(i = 0\) até \(m\):

\(P = \text{Deg a} \ circ \text{polo}\ (j, m-j)\) \(\mathcal{O}(c)\)

Se \(C[j: P]\) = \(\text{Reverse}\ (C[P: m-j])\) \(\mathcal{O}(m/2)\)

Return \(j, m-j\)