If true, \( \# a \# \# 1 \# \# \) we return it by \( m^3 \) which is \( O(m^0) \).

If false, we increase \( m \) value by 1.

Then we get the state

\[ \# a \# 1 \# \# \]

Now we copy again:

\[ \# a \# 1 \# \# a \# 1 \# \# \]

If true, \[ \text{apply } P(a, 1) \]

If false:

\[ \# a \# 1 \# 1 \# \]

Return 1. Increase \( m \) by 1. Then copy & repeat the steps.

Finally we will get \( \lambda \) for which \( P(a, m) \) is true.

Next Thursday - test.

**Big picture:**

\( \mu \)-recursive fn

\( (\text{church}) \)

\( \text{TM (Turing)} \)

Proving that every \( \mu \)-rec \( f \) is Turing-computable.

\( \mu m \ P(\overline{a} \ , m) = f(\overline{a}) \)

\( \overline{a} = (a_1 \ldots a_n) \)

We know that \( P(\overline{a} \ , m) \) is Turing computable.

\( \text{we want:} \)

\[ \# \overline{a} \# \# \# \]

We start with

\[ \# \overline{a} \# \# \# \# \]

We analyze the problem:

\text{Idea:} try \( m = 0 \); if \( P(\overline{a} \ , 0) \) return 0, else try \( m = 1 \).

\( \text{if } P(\overline{a} \ , 1) \) return 1.

\[ m = 0; \]

\[ \text{while} (P(\overline{a} \ , m)) \]

If we start with \( \# \overline{a} \# \# \# \) if \( P(\overline{a} \ , 0) \) false.

\[ \text{Then } \# \# \# \# \]

We are left.

Correct way:

Step 1:

\[ \# a \# 1 \# \]

Copy

\[ \# a \# 1 \# a \# 1 \# \# \]
we return 0 by
\[ \pi_{k+2} \] i.e. \( \pi_2 \) here.

\[ \# \alpha \# \# 1 \# \#
\]

we go right & see if
\[ \# \bar{\alpha} \# \# 1 \#
\]

means false.

\[ \text{copy again.} \]

2 possibilities, \( p(\bar{\alpha}, 1), p(\bar{\alpha}, 1) \) fail.

\[ \# \alpha \# \# 1 \# \#
\]

\[ \text{true} \]

\[ \# \bar{\alpha} \# \# 1 \# \#
\]

\[ \# \bar{\alpha} \# \# 1 \# \#
\]

\[ \text{return } m = 1 \text{ by } \pi_{k+2} \]

\[ \text{increase by } 1. \]

\[ \# \bar{\alpha} \# \# 1 \# \#
\]

then again copy \( \pi \)

\[ \# \bar{\alpha} \# \# 1 \# \#
\]

\[ \# \bar{\alpha} \# \# 1 \# \#
\]

\[ \text{True finally we get, } m \text{ for which } \#
\]

\[ \text{\( p(\bar{\alpha}, m) \) is true.} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]

\[ \text{\# \# \# \#} \]
How do we define an algo?

1930s set TM

\( \mu \)-recursion

How do we define a feasible algorithm? No perfect definition exists.

**Existing definition:**
- Sorting \( O(n \log n) \)
- Exhaustive search \( 2^n \) not feasible
- Linear search \( O(n) \)
- Matrix multiplication \( O(n^3) \)

**A little bit of physics:**

\( T \approx 20 \text{ billion years} = 2 \times 10^{10} \text{ years} \) (timeline of universe)

we can make, \( \frac{T}{\text{computational steps}} \rightarrow \text{smallest possible time interval} \)

\[ 1 \text{ year} = 365 \text{ days} \approx 4 \times 10^8 \text{ days} = 4 \times 10^2 \times 2.10^1 \times 6.10^4 \]

\[ 1 \text{ day} = 24 \text{ hrs} \approx 2.10^1 \text{ hrs} = 300 \times 10^5 \approx 3 \times 10^7 \text{ secs} \]

\[ 1 \text{ hr} = 60 \text{ min} \approx 6.10^1 \text{ min} \]

\[ 1 \text{ min} = 60 \text{ sec} = 6.10^1 \text{ sec} \]

\[ 1 \text{ THz} = 10^9 \text{ operations} \]

\[ 1 \text{ THz} = 10^{12} \text{ operations} \]

\[ 1 \text{ PHz} = 10^{15} \text{ operations} \]

\[ 1 \text{ year} = 3.15 \times 10^{35} \text{ steps} \]

\[ \Delta t = \frac{\text{dist}}{\text{velo}} = \frac{c}{\text{speed of light}} \]

\( c = 3 \times 10^8 \text{ m/s} \)

\( \Delta t \) time during which the light goes through the smallest elementary particle.

\( \text{Heisenberg's Uncertainty Principle} \)

\( \Delta x \cdot \Delta p \geq \hbar \)

\( \hbar = \text{Planck's constant} \)

\( \mu \text{B} \)
So, \( \Delta x > \frac{\hbar}{m} \) \( \rightarrow \) velocity of particle.

Proton's size is smallest.

\[ \Delta t = \frac{\text{size of a proton}}{c} \approx 10^{-23} \text{ secs} \]

\[ \Delta x \approx 10^{-13} \text{ cm}, \quad c = 3 \times 10^5 \text{ km/sec} \]

Thus, computational steps that can be done \( \rightarrow \frac{6 \times 10^{17}}{10^{23}} \approx 6 \times 10^{40} \)

during \( T \)

\( 10^{90} \) normal particles. If we consider all the particles are \( 10^{90} \) particles overall computing, then \( 10^{120} \) steps

So, \( 26^n = 10^{130} \). \( \left(10^{15}\right)^n \approx 10^{130} \)

\[ n = \frac{130}{15} \approx 8.7 \]

So, [input size 84 for exhaustive search]