Chapter 7. Time Complexity

Def Let M be a deterministic TM that halts on all inputs. The running time of M is a function $f: N \rightarrow N$, where $f(n)$ is the maximum # of steps that M uses on any input of length n.

Def $f(n)$ is $O(g(n))$ if there exists constant c, and integer n_0 such that, for $n \geq n_0$,

$$f(n) \leq c \cdot g(n).$$

Intuitively, $g(n)$ is an upper bound for $f(n)$ — up to some constant factor when n is large.

Exercise.

- $f_1(n) = 3n^2$, $g_1(n) = n^3$
- $f_2(n) = 3n^2$, $g_2(n) = n^2$

Is $f_i(n) = O(g_i(n))$? why?

Try to spend 10 minutes.

<u>Answer</u>: ① $f_1(n) = O(g_1(n))$ as we can find $C=3$, $n_0=1$ such that

$f_1(n) = 3n^2 \leq 3 \cdot n^3$, when $n \geq 1$.

② $f_2(n) = O(g_2(n))$ as we can find $C=3$, $n_0=1$ such that

$f_2(n) = 3n^2 \leq 3 \cdot n^2$, when $n \geq 1$.

Note: $f_2(n) > g_2(n)$ when $n \geq 1$.

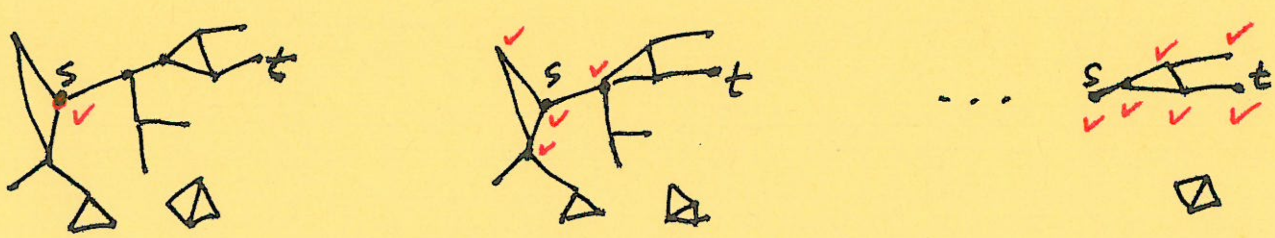
> In reality, when one uses O-notation, usually one has to make the bound (e.g. $g_1(n)$) as small as possible.

- <u>Def</u> $\mathrm{TIME}(t(n)) = \{ L \mid L$ is a language decided by an $O(t(n))$ time TM$\}$.

<u>Example</u>

PATH $= \{ \langle G, s, t \rangle \mid G$ is a directed graph that has a directed path from s to $t\}$.

n — # of vertices in G.



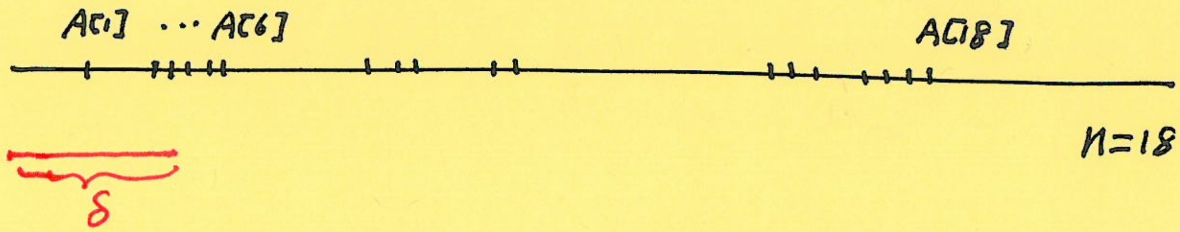
$\therefore$ PATH $\in$ TIME(n^2).

<u>one more example</u>:

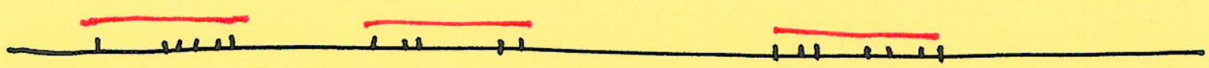
Given an array of sorted real numbers, and an interval of length δ, use the minimum # of such intervals to cover these real numbers

$A[1], A[2], \cdots, A[n]$.

input:



output:



So, opt = 3.

<u>requirement</u>: Solve this problem as fast as you can. We'll cover the solution later.

<u>CLASS P</u>:

$$P = \bigcup_k \text{TIME}(n^k)$$

P — all problems which can be solved in polynomial time.

For instance, Sorting $\in$ TIME$(n \log n)$

linear search $\in$ TIME(n).