Time Complexity or "Running Time"

Let M be a TM. Let $f: \mathbb{N} \to \mathbb{N}$ be a function. We say that M has Time Complexity $f(n)$ if:

$$\forall x, \quad |x| = n$$

M on input x halts in at most $f(n)$ steps.

The exact running time of an algorithm is often a complex expression. Usually estimated with asymptotic analysis.

we consider only the highest order term. Disregarding both the coefficient of that term and any lower order terms.

Let f and g be functions

$$f, g : \mathbb{N} \rightarrow \mathbb{R}^+$$

We say that $f(n) = O(g(n))$ if positive integers c and n_0 exist such that for every integer $n \geq n_0$

$$f(n) \leq c\,g(n)$$

When $f(n) = O(g(n))$ we say that $g(n)$ is an asymptotic upper bound for $f(n)$.

Big-O notatation represents that we are throwing away constant factors

example

$$f(n) = 7n^3 + 4n^2 + 3n + 1000 \qquad O(n^3)$$

example

$$f(n) = 3\log_{10} n + 2\log_{10}\log_2 \cancel{A} + 1$$

$$O(\log n) \qquad \log_b n = \frac{\log_2(n)}{\log_2(b)}$$

Time-complexity classes:

Let $t: N \to R^+$ be a function. Define the time-complexity class, $TIME(t(n))$, to be the collection of all languages that are decidable by an $O(t(n))$ time Turing-Machine.

$$A = \{0^k 1^k \mid k \geq 0\}$$

M_1 = "On input w:

1. Scan accross tape to verify it is of the form $0^* 1^*$, if not REJECT
2. Zigzag accross the tape crossing off a 0 and a 1 each time untill all 0s or all 1s are all crossed off.
3. If there exists unmarked 0s or 1s REJECT otherwise ACCEPT."

What is the time-complexity class of A?

Using M_1 $A \in TIME(n^2)$

M_2 = "On input w:

1. Scan across the tape and REJECT if a 0 is found to the right of a 1
2. Repeat as long as some 0s and some 1s remain
3. Scan across the tape, checking whether the total number of 0s and 1s remaining is even or odd. If odd REJECT
4. Scan across the tape, crossing off every other 0 starting with the first. Then crossing off every other 1 starting with the first.
5. If no 0s and no 1s remain on the tape ACCEPT. Otherwise REJECT"

$A \in \mathrm{TIME}(n \log n)$

M_3 = "On input w:

1. Scan accross the input tape and reject if a 0 follows a 1
2. Scan across the 0s on the input tape Copying the 0s onto the working tape.
3. Scan accross the 1s on the input tape. For each 1 cross off a 0 on the working tape. If all 0s are crossed off before all 1s are read REJECT.
4. If all 0s are crossed off ACCEPT. If any remain REJECT."

$A \in TIME(n)$

Consider time complexity for

$$\{0^k 1^k \mid k \geq 0\}$$

M = " On input w

1. Scan accross the input to ensure it is in the form $0^* 1^*$ $O(n)$

2. Zig-zag accross the tape crossing off a 0 and a 1 each time until 0s or 1s are all crossed off. $O(n^2)$

3. if there exists unmarked 0s or 1s REJECT $O(n)$

otherwise ACCEPT "

Multi-tape TM

If $t(n) \geq n$ then for every $t(n)$ - time MT TM, $\exists$ a $O(t^2(n))$ TM

- works for all deterministic variants of TM

Multi-tape TM

If $t(n) \geq n$ then for every $t(n)$-time MT TM $\exists$ a $O(t^2(n))$ time single tape TM.

Proof idea:

Use the proof of equivalence between MT TM and single-tape TM. Show that simulating each step of a multi-tape machine uses at most $O(t(n))$ steps on the single tape machine.

Proof

- Let M be a k-tape TM that runs in $t(n)$ time.
- Construct single-tape TM S that runs in $O(t^2(n))$ time.
  - S uses a single tape to represent all k-tapes
  - tapes are stored consecutively, with the positions of the tape heads marked.
- For each step of M, S makes 2 passes over its tape.
- Each of M's k tapes has an active portion at most $t(n)$ in length
- S performs 2 scans and possibly k rightward shifts.
- each scan uses $O(t(n))$ time

  Initial setup $- O(n)$

  each step $- O(t(n))$

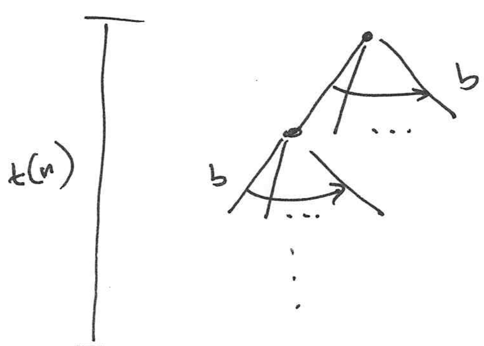
  number of steps $- t(n)$

total time $O(n) + O(t^2(n))$

since $t(n) \geq n$ $O(t^2(n))$

Let N be a non-deterministic TM that is a decider (i.e. halts on every input). The running time of N is the function $t : \mathbb{N} \to \mathbb{N}$ where $t(n)$ is the max number of steps on any branch of computation on any input of length n

<u>Claim</u>: Let $t(n) \geq n$ then every $t(n)$-time non-deterministic TM has an equivalent $2^{O(t(n))}$ - time MT TM



$$\# \text{ of rounds} \leq 1 + b + b^2 + \dots + b^t = \frac{b^{t+1} - 1}{b - 1}$$
$$= c(b^{t+1} - 1)$$
$$= O(b^{t+1})$$

$$\# \text{ of steps} \leq t\, O(b^{t+1})$$
$$= O(t\, b^{t+1})$$
$$= O(b^{\log_b t + t})$$
$$= O(b^{2t})$$
$$= b^{O(t(n))}$$
$$= 2^{O(t(n))}$$