Space complexity

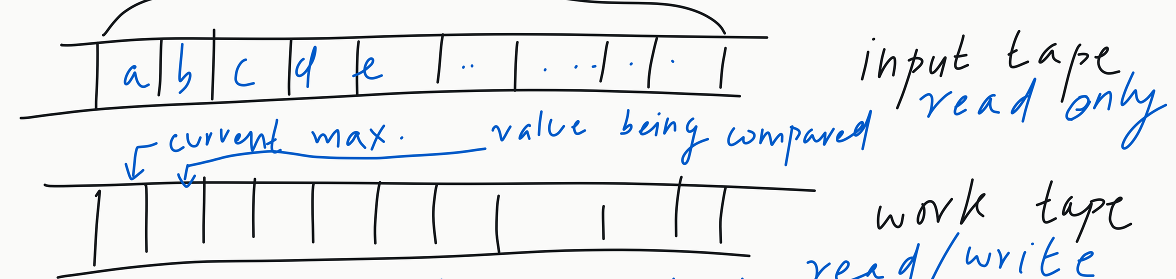
SPACE$(s(n))$, $s: \mathbb{N} \to \mathbb{N}$

finding the maximum in a sequence

SPACE (1)

O(1) space

n numbers



cells required on work-tape

SPACE(s(n)) $s(n) \geq \log n$

Is there a relation b/w ~~length~~ size of sequence and the binary representation of a number in the sequence?

No

SPACE $(s(n))$ NSPACE$(s(n))$

non deterministic TM

- input, work tape(s)
- standard NDTM.

$\boxed{DTIME(s(n)) \subseteq SPACE(s(n))}$

perform every computation on a new cell, it would take SPACE $(s(n))$

SPACE $(s(n)) \subseteq$ NSPACE $(s(n))$

$\boxed{NSPACE(s(n)) \subseteq DTIME(2^{O(s(n))})}$

$s(n)$ # of cells

Configuration Graph (of a TM):

① A C.G. consists of all the contents of all tapes, locations of all pointers, at a particular point in a T.M's execution.

② $G_{M,x}$ is a directed graph whose nodes correspond to possible configurations that M can reach from the starting configuration $C_{x,start}$ (where the input tape is initialized to contain x).

M is TM, x is input

③ The graph has a directed edge from a configuration C to a configuration C', if C' can be reached in one step from C.

④ If M is deterministic, the graph has a max. out degree of 1.

If M is non deterministic, the graph has a max. out degree of ~~b~~ 2, given that M branches for almost b branches at any point.

⑤ M's computation doesn't repeat the same configuration twice (as otherwise it will enter an infinite loop).

i.e. it is a DAG.

⑥ M accepts x iff there exists a path from $C_{x,start}$ to $C_{x,accept}$

How many nodes does a C.G. have. (upper bound)?

hint: each node:

$O(s(n))$ cells ← ① entire contents of all tapes

$< O(s(n))$ ← ② all pointers.

each node can be described in $O(s(n))$ bits.

$2^{O(s(n))}$ possible nodes

∴ the C.G

Run a graph traversal on the C.G. with source: $C_{x,start}$ and target: $C_{x,accept}$ and check whether a path exists.

takes time linear to # nodes.

∴ $\boxed{NSPACE(s(n)) \subseteq DTIME(2^{O(s(n))})}$

$\boxed{DTIME(s(n)) \subseteq SPACE(s(n)) \subseteq NSPACE(s(n)) \subseteq DTIME(2^{O(s(n))})}$ → e.g. → 2-player games.

PSPACE $\subseteq$ NPSPACE $\subseteq$ EXPTIME

NP ? $\subseteq$ EXPTIME

puzzles

$$PSPACE = \bigcup_{c>0} SPACE(n^c)$$

$$NPSPACE = \bigcup_{c>0} NSPACE(n^c)$$

$\boxed{3SAT \in PSPACE?}$

Space required: → n cells for variables

→ a constant # cells to check clause by clause

∴ $3SAT \in SPACE(n) \in PSPACE$