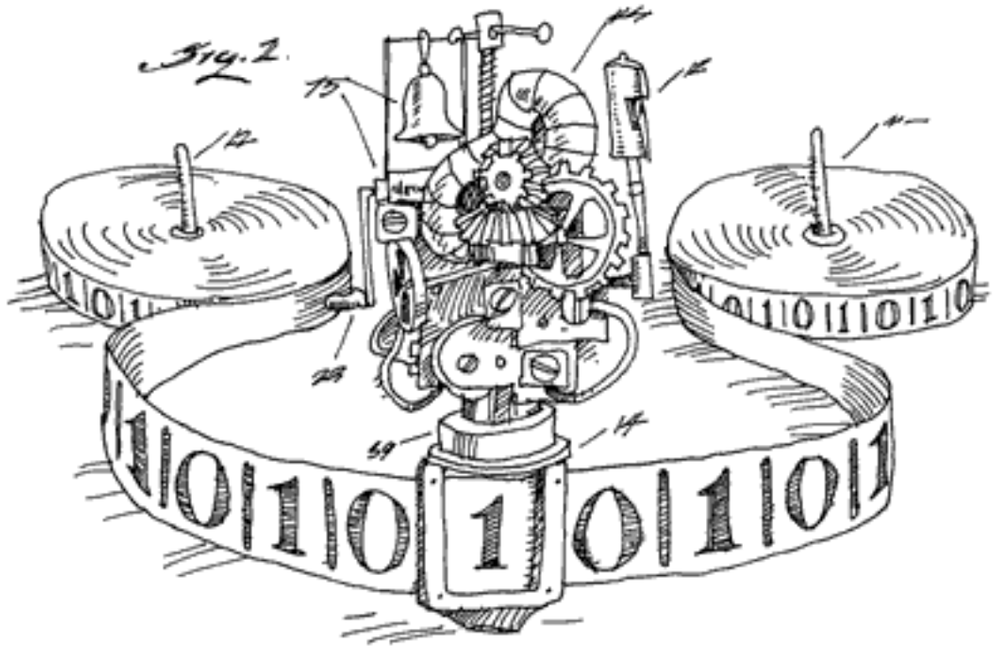
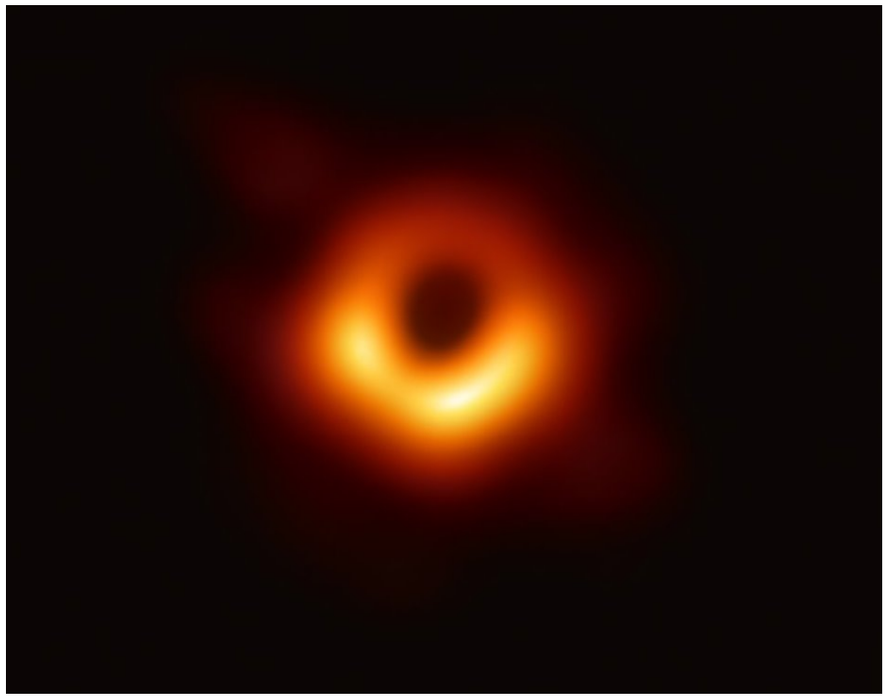
CSC4010: Introduction to Theoretical Computer Science



Lecture 18

Space Complexity





Left: The first reconstructed image of a black hole emerge from the Event Horizon Telescope data.

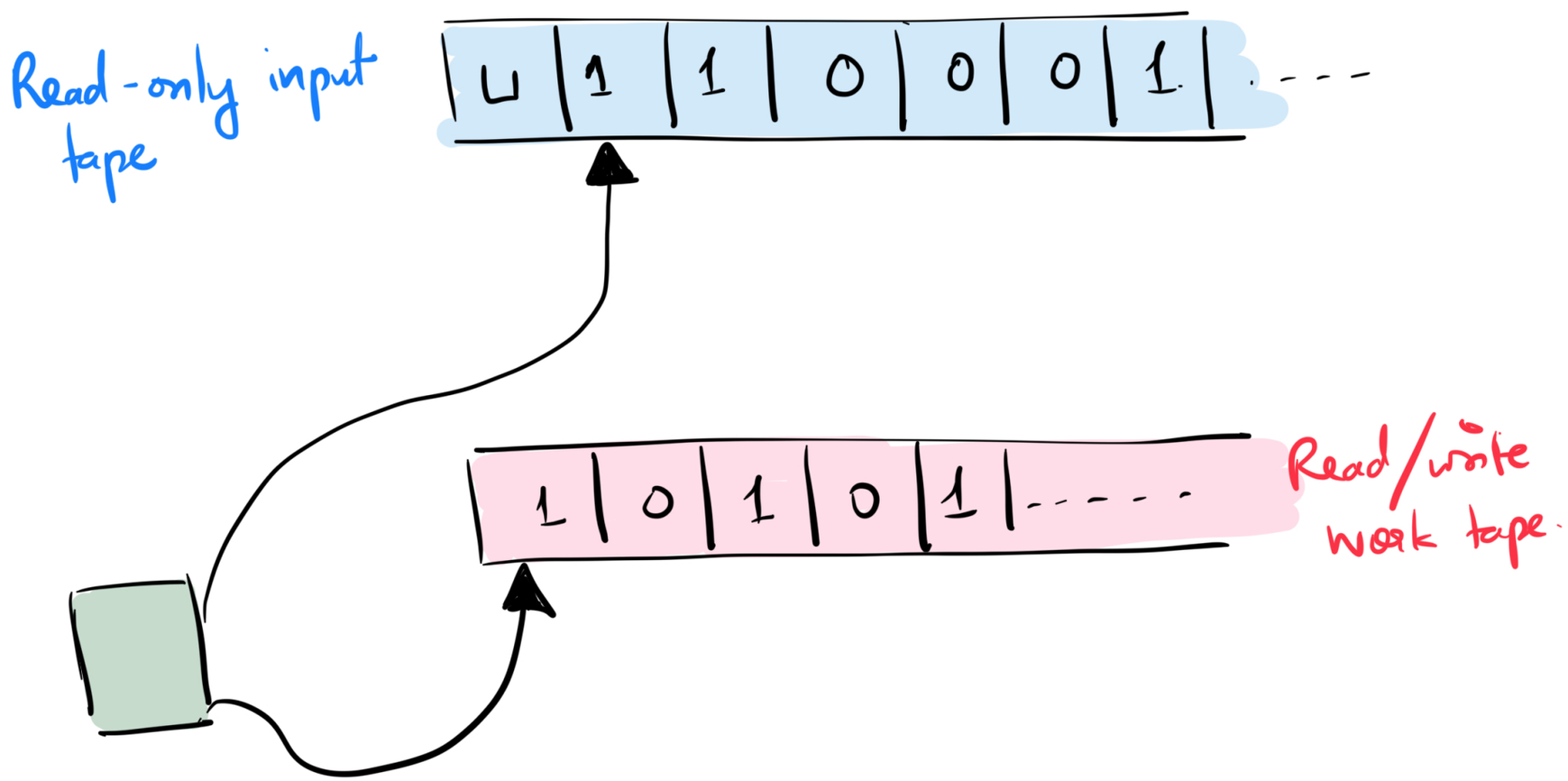
Right: Dr. Katie Bouman, imaging scientist, posing near hard drives containing some of the **petabytes of black hole imaging data** collected by telescopes around the world.

- The capacity of a device's readily accessible memory, such as RAM, is only a few gigabytes!

Space Complexity

Can we solve any interesting problem in $o(n)$ space?

The one-tape Turing machine is not the right model of computation for studying sub linear-space algorithms.



Space Complexity: Complexity class SPACE(s)

Let L be a language and let $s : \mathbb{N} \to \mathbb{N}$ be a (space bound)

Definition: $L \in \text{SPACE}(s)$ if there is a two-tape Turing machine M such that:

- M decides L
- M never modifies the symbols written on tape 1
- We have $s_M(n) = O(s(n))$, where $s_M(n)$ is the maximum i such that the tape 2 head visits cell i during the computation of M on w for some $w \in \{0,1\}^n$.

$\text{SPACE}(s) = \{L \subseteq \{0,1\}^*$ that can be decided by a TM with worst-case space $O(s)\}$

Complexity Classes

PSPACE $= \cup_{k \geq 1}$ SPACE(n^k)

L $=$ SPACE$(\log n)$

L is the set of languages that can be decided in logarithmic space.

EXPSPACE $= \cup_{k > 1}$ SPACE(2^{n^k})

BALANCED $\in L$

BALANCED = $\{x \in \{0,1\}^* : x$ has equal number of zeroes and ones$\}$

Claim: BALANCED $\in L$

Proof Sketch: Given $x \in \{0,1\}^n$:

- Count the number of ones in x
- Count the number of zeroes in x
- Check if the two counts are equal

SAT ∈ PSPACE

Corollary: NP ⊆ PSPACE

Time-Space Hierarchy

Theorem: For every function $s : \mathbb{N} \to \mathbb{N}$, SPACE($s$) $\subseteq$ TIME($2^{O(s)}$)

Space Hierarchy Theorem

Theorem: Let s be a space-constructible function. Then, $\text{SPACE}(s) \subsetneq \text{SPACE}(s \log s)$.

Space Hierarchy Theorem

Theorem: Let s be a space-constructible function. Then, $\text{SPACE}(s) \subsetneq \text{SPACE}(s \log s)$.

L vs. P vs. NP vs. PSPACE

$L \subseteq P \subseteq NP \subseteq PSPACE$

What we expect: **All** of these containments are **strict.**

What we can prove: **At least one** of these containments is **strict.**

Theorem: $L \neq PSPACE$

The L vs. P problem

We expect that $L \neq P$, but we don't know how to prove it.

$L = P$ would mean that every efficient algorithm can be modified so that it uses a tiny amount of workspace.

Class NL

NSPACE(s) stands for non-deterministic space s .

NSPACE(s) is the class of languages for which membership can be verified in space $o(s)$ and the verifier can read the certificate one time from left to right.

NL stands for non-deterministic logspace.

This is the class of languages for which membership can be verified in logarithmic space - with the additional requirement that the verifier can only read the certificate one time from left to right.

STCONN $\in$ SPACE$(k \log n)$

STCONN $= \{\langle G, s, t, k\rangle : G$ is a directed graph, there is a path of length $\leq k$ from s to t in $G\}$

STCONN $\in$ NL

STCONN $= \{\langle G, s, t, k\rangle : G$ is a directed graph, there is a path of length $\leq k$ from s to t in $G\}$

Theorem: STCONN $\in$ NL

Next Time

- There is a TM that decides STCONN in space $O(\log k \log n)$.

- **Savitch's Theorem:** For every $s : \mathbb{N} \to \mathbb{N}$, NSPACE$(s(n)) \subseteq$ SPACE$(s(n)^2)$

- PSPACE = NPSPACE