3s. Complexity Classes

Monday, February 14, 2022 9:02 PM

Candidate
Thurs: Research talk: 4:30 in 224
Friday: Mock class 9:00 am 202
Friday: 2:35 – 3:30 p.m. Open House with students 75 SHS 2nd Floor East Lounge

Goals

- Understand how to do Pset Assessment
- Define DTIME, P, and EXP, our first complexity classes!
- Discuss real world connection

def: DTIME

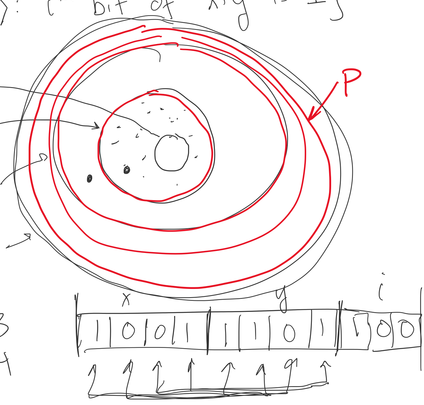
Let $T: \mathbb{N} \to \mathbb{N}$. A Language $L \subseteq \{0,1\}^*$ is in $DTIME(T(n))$ iff $\exists$ a TM M that decides L in $O(T(n))$ steps.

ex: $DTIME(O(n \log n))$

What complexity class is

$L = \{\langle x, y, i \rangle : i^{th}$ bit of $x \times y$ is $1\}$

~~A)~~ $DTIME(\log(n))$
B) $DTIME(n)$
C) $DTIME(n^2)$
D) $DTIME(2^n)$



ex: $x = 9$, $y = 13$, $i = 4$

Big-Picture: Addition is easy

def: P ("Polynomial-Time")

$$P = \bigcup_{d \geq 1} DTIME(n^d)$$

We say P characterizes efficiently computable (easy) problems.

Good? Bad? Issues? (With calling P efficiently computable problems?)

- "Easy" is subjective → what can do in head (can do hard things)
  - n^{1000} is not easy (Paul)
- No good cut-off → this is best option (good enuf)
  b/c there is a clear cutoff
- Only captures what comp. do, not humans/animals
  not natural alg.
  - No space limits
  - Quantum
  - Randomness

def: EXP ("Exponential Time")

$$EXP = \bigcup_{d \geq 1} DTIME(2^{n^d})$$

Hard/
Not easy problems