While we wait...

- Are there problems that we can solve in $O(n^2)$ but not $O(n)$?
- What about $O(n^3)$ but not $O(n^2)$?
- What about $O(n^{100})$ but not $O(n^{99})$?
- What about $O(n^{1.00001})$ but not $O(n)$?
- What about $O(n \cdot \log(\log(n)))$ but not $O(n)$?
- What are some resources other than time that are useful in computation.

CS 463 Tutorial 9: Zooming in on P + a new resource

TA: Harry Sha (shaharry@cs.toronto.edu)

March 16th, 2022

Time is precious

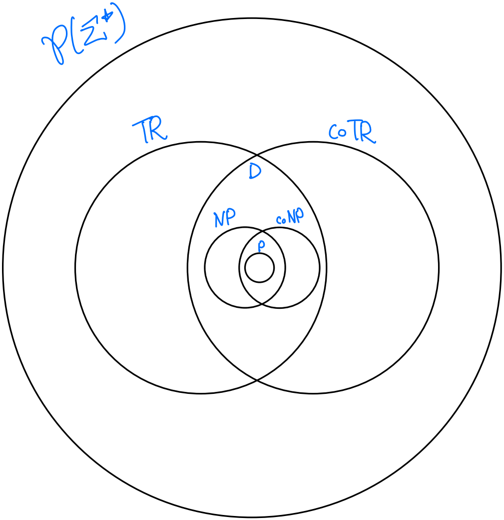
The big question for the first part of today is:

> Can I decide strictly more problems given more time, and how much more time do I need?

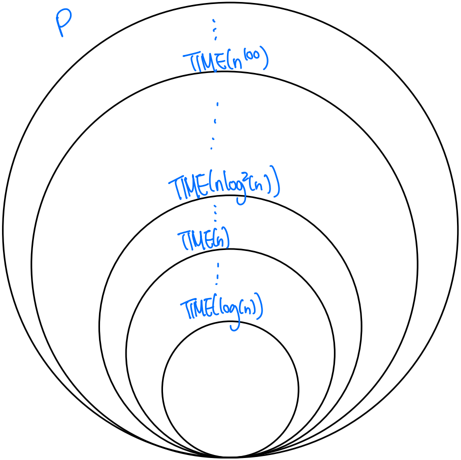
Let $\mathrm{TIME}(f(n))$ be the class of decision problems that can be solved in $O(f(n))$ time on a (deterministic) TM.

Note that using this notation, $\mathrm{P} = \bigcup_{c \geq 0} \mathrm{TIME}(n^c)$.

The picture



Zoomed in



Time Hierarchy Theorem

The answer to our question is yes (sort of).

Theorem (Time Hierarchy Theorem)

If f, g are functions such that $f(n)\log(f(n)) = o(g(n))$. Then

$$\mathrm{TIME}(f(n)) \subsetneq \mathrm{TIME}(g(n))$$

Time Hierarchy Theorem

Theorem (Time Hierarchy Theorem)

If f, g are functions such that $f(n)\log(f(n)) = o(g(n))$. Then

$$\mathrm{TIME}(f(n)) \subsetneq \mathrm{TIME}(g(n))$$

For example, this shows $\mathrm{TIME}(n^9) \subsetneq \mathrm{TIME}(n^{10})$ since

$$n^9 \log(n^9) = 9n^9 \log(n) = o(n^{10})$$

Proof of the Time Hierarchy Theorem

Two lemmas

Lemma (Nice representation of TMs)

There is a way to represent TMs such that

- *Every string in Σ^* corresponds to some TM.*
- *Every TM is represented infinitely many times.*

Lemma (Efficient universal TM)

There is a universal TM U that simulates any TM M on any input x such that if M runs for T steps on x, the simulation runs for $CT\log(T)$ steps.

Proof of the Time Hierarchy Theorem

The plan is to use diagonalization to find some language L s.t. $L \in \mathrm{TIME}(g(n))$ but $L \notin \mathrm{TIME}(f(n))$.

* steps on $\mathcal{U}$, not M_w!

$D(w)$:

run M_w on w using the universal TM for $g(|w|)$ steps.*
if M_w halts:
    return the flipped result.

* else case not important

Let $L = L(D)$. Note D runs in time $O(g(n))$. By contradiction, assume there is some $O(f(n))$ decider M s.t. $L(M) = L$

Proof of the Time Hierarchy Theorem

For any input of length n, M runs for at most $cf(n)$ steps. On the universal TM, this takes at most $c'f(n)\log(f(n))$ steps. Since $f(n)\log(f(n)) \in o(g(n))$ by assumption, for large enough n, $g(n) > c'f(n)\log(f(n))$.

Pick some α such that $|\alpha| > n$, and $M_\alpha \equiv M$. One exists b/c of lemma 1. Now consider running D on α.

Since $g(|\alpha|) > c'f(n)\log(f(n))$, the simulation halts. Then $L(D) \neq L(M_\alpha)$ since α is in exactly one of them. But this is a contradiction

Corrolaries

- $\mathrm{TIME}(n^k) \subsetneq \mathrm{TIME}(n^{k+\epsilon})$ for any $k \geq 0$, $\epsilon > 0$
- $P \subsetneq \mathrm{TIME}(2^n)$

What are some other useful resources for computation?

Randomness

What are some uses of randomness?

Some examples

- Quicksort
- Find an 1 in an length n array with half 1s and half 0s.

Probabilistic TMs

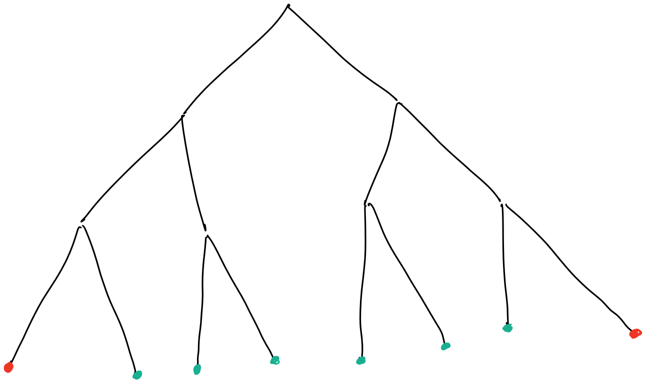
Like non-deterministic TMs, except the branching factor is at most 2. When the execution hits a non-deterministic step, the TM flips a fair coin to decide which path to follow.

Note that a branch of height k is taken with probability 2^{-k}.

The probability that a TM, M, accepts w is

$$\sum_{b, b \text{ is an accepting branch}} \Pr[b]$$

Picture



Errors

	$x \in L$	$x \notin L$
M accepts x	True positive	False positive
M rejects x	False negative	True negative

Complexity Classes

Let BPP (bounded-error probabilistic polynomial time) be the set of problems L for which there exists a polynomial time probabilistic TM such that for all $x \in \Sigma^*$, the TM errs on x with probability most $1/3$.

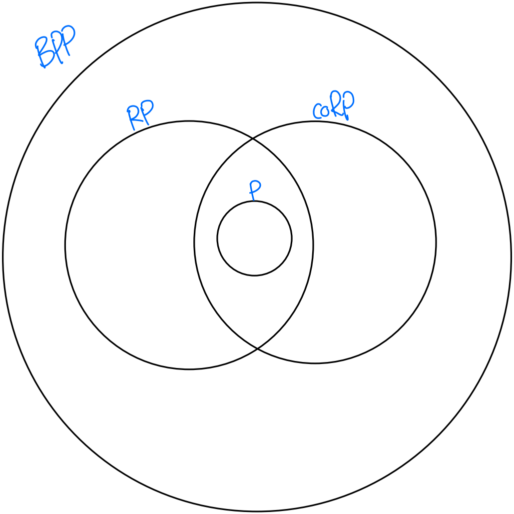
- $\mathrm{RP} \subset \mathrm{BPP}$ is the subset that doesn't allow false positives. I.e. for every $x \notin L$, the TM rejects with probability 1.
- $\mathrm{coRP} \subset \mathrm{BPP}$ is the subset that doesn't allow false negatives. I.e. for every $x \in L$, the TM accepts with probability 1.

Comparison

Class	$x \in L$	$x \notin L$	
BPP	$\geq 2/3$	$\leq 1/3$	
RP	$\geq 2/3$	$= 0$	
coRP	$= 1$	$\leq 1/3$	
NP	> 0	$= 0$	

Accept probability

Picture



FAQ

1. How does randomness compare with non-determinism? I.e. what is the relationship between RP and NP or BPP and NP?
2. Does having error on both sides help? I.e. is $\mathrm{RP} = \mathrm{BPP}$?
3. What if we relax the requirement of polytime to expected polynomial time?
4. Can we buy accuracy with more randomness and time?
5. Does randomness actually help? I.e. does $\mathrm{BPP} = \mathrm{P}$? Or how much time does randomness cost?

FAQ answers

1. $\mathrm{RP} \subset \mathrm{NP}$ (see the comparison slide and observe that a RP decider is a NP decider), but the relationship between BPP and NP is unknown. I.e. we don't know if $\mathrm{BPP} \subseteq \mathrm{NP}$ or the other way around or both.
2. Unknown!
3. We can relax to this definiton and the classes don't change!
4. Yes! Run the algorithm independently many times and output the majority answer
5. Unknown! But, surprisingly, currently people believe everything can be derandomized i.e. $\mathrm{BPP} = \mathrm{P}$!

There a lot unknown about randomized complexity classes. For all we know right now, it might be the case that $\mathrm{BPP} = \mathrm{EXP}$!