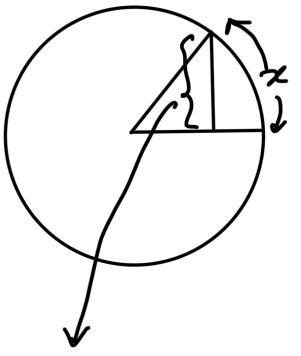
COL 351 Lecture 1 2023/01/04

Agenda:

- Overview of the course
- The boring logistics
- Some fun

What came first? Computers or Algorithms?

[Madhava, 14th century] Sine computation algo.



$T_0 = x$

Obtain the ith T_i term by multiplying T_{i-1} by x^2 and dividing by $(2i)^2 + 2i$

T_0

T_1

T_2

T_3

T_4

T_{n-2} $T_{n-2} - (T_{n-1} - T_n)$

~~T_{n-1}~~ $T_{n-1} - T_n$

~~T_n~~

$$T_0 - (T_1 - (T_2 - (\ldots\ldots (T_{n-1} - T_n))\ldots))$$

$$T_n = \frac{x^{2n+1}}{(2n+1)!}$$

$$T_0 - T_1 + T_2 - T_3 \ldots\ldots$$

Course objectives:

1. How to design algorithms
   (Designing an algorithm ≠ knowing an algorithm)
2. How to write algorithms in a human language
   – while still being clear and unambiguous
3. How to show the non-existence* of efficient algorithms

(*): "If there is an efficient algorithm for this problem, then there are efficient algorithms for tonnes of other problems. However, no such algorithm is known to the humanity for any one of those tonnes of problems."

Example: Quicksort – Algorithmic idea:

(How to write algorithms)

Given array A of items from an ordered set:

- If $|A| \leq 1$, return.
- Pick one item, say p (a.k.a. pivot)
- A_1: items in A that are $< p$
- A_2: items in A that are $> p$
- Apply Quicksort recursively to A_1, A_2.
- A ← 

Example: Quicksort — Implementation Details (mostly unnecessary)

(How NOT to write algorithms)

"Pick the first element as pivot. Starting from the second element, run pointer i forward until you find an element > pivot. Starting from the last element, run pointer j backward until you find an element < pivot. Exchange the ith and jth element, and repeat until i and j cross. Now exchange the pivot with the jth element and recurse ..... (more pain in the *** for the reader)"

Coming up next:

Inter — ~~department~~ entry-year

Data Structures Quiz

Team 1: 2019 entry

Team 2: 2020 entry

Team 3: Everyone else

TODO: Each team elects 3 members for the quiz, to be held in the next class.