CS 120: Intro to Algorithms and their Limitations

Lecture 3: Reduction — Thursday, September 7, 2023

Pset Due: September 13, 2023 **Denny Cao**

---

§1 Comparison-Based Sorting

- Exhaustive Search
- Insertion Sort
- Merge Sort

Proof of Lower Bound. The input is: $((\sigma(0), \emptyset), \ldots, (\sigma(n), \emptyset)$.
The output is $((0, \emptyset), \ldots, (n, \emptyset))$.
$2^T > n!$. Represents number of possible outcomes (permutations) will be at least $n!$. We take the log of both sides $= \Omega\left(\frac{n}{e}^n\right)$ □

> **Remark 1.1.** Look over lecture notes... Don't understand

§2 Reductions

§2.1 Duplicate Search

Problem 2.1 (Duplicate Search Problem)**.** Input: $a_0, \ldots, a_{n-1}$
Output: A duplicate number; such that $\exists i \neq j \mid a_i = a_j = a$

> **Algorithm 2.2** ($O(n^2)$: Naive Approach) — For each a_i, loop through all a_j, $i < j \leq n-1$. Return a_i if duplicate found else $\perp$.

> **Algorithm 2.3** ($O(n \log n)$: Sorted Array Approach) —
>
> 1. Form an array $(k_0, v_0), \ldots, (k_{n-1}, v_{n-1})$ with $k_i = a_i, v_i = \emptyset$: Output S
> 2. Sort
> 3. For $0 \leq i < n-1$, check if $S[i] = S[i+1]$

Proof of Correctness. Check lecture notes. □

> **Remark 2.4.** By convention, sorting problems are for key-value pairs. $v_i = \emptyset$ because we are ignoring the value and just sorting keys.

We will formalize reduction:

Definition 2.5 (Formalism of Reduction)**.** Input $\Pi = (\mathcal{I}, \mathcal{O}, f)$
Oracle $\Gamma = (J, \varphi, g)$.

Definition 2.6 (Oracle)**.** An *oracle* is a function which given $x \in \mathcal{J}$, outputs on element of $g(x)$.

Definition 2.7 (Reduction)**.** A *reduction* is an algorithm that solves Π using an oracle or Γ as a subroutine.

Remark 2.8. $\Pi \leq \Gamma$

Remark 2.9. For duplication search algorithm 2.3, oracle is the sorting algorithm. We don't have to prove the oracle again. Do not need to be specific about oracle; can just say "sort".