More on Computation Models (Slides for ESTR2102)

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong

A Remark on the RAM Model

In our definition of the RAM model, the CPU is allowed to do only these atomic operations: (i) register initialization, (ii) arithmetics, (iii) comparisons, and (iv) memory access.

In another popular version of the RAM model—dubbed as RAM2 in this lecture—the CPU is also endowed with the power of logic computation. Specifically, it is allowed to do two more atomic operations:

- `AND`: given two registers a and b, compute a `AND` b (namely, taking the "AND" of the two words stored in a, b), and store the result in a designated register.
- $\neg$: given a register a, compute $\neg a$ (namely, flipping each bit of the word in a), and store the result in a designated register.

A Remark on the RAM Model

In the RAM2 model, a OR b can be computed in $O(1)$ time:

$$a \texttt{ OR } b = \neg(\neg a \texttt{ AND } \neg b).$$

So can a XOR b:

$$a \texttt{ XOR } b = (a \texttt{ OR } b) \texttt{ AND } \neg(a \texttt{ AND } b).$$

A Remark on the RAM Model

Think: How to implement `AND` in the RAM model we defined in the class? You may use $O(w)$ time, where w is the number of bits in a word.

Discuss: What makes a good computation model? In particular:

- Should $\lfloor \sqrt{n} \rfloor$ be an atomic operation?
- Should we remove multiplication as an atomic operation (note that multiplication can be implemented using addition)?
- Should we go for RAM2, as opposed to RAM, since the former is more powerful?

A Weaker Model – The Comparison Model

Let us re-visit the dictionary search problem discussed in the class.

Recall that in this problem, we are given a set S of n integers, already arranged in ascending order. We are given a search value q, and must decide whether $q \in S$.

We discussed in the lecture two algorithms: sequential scan and binary search. Our discussion was in the RAM model, which (interestingly) is "too powerful" for studying the essence of the two algorithms. Next, we will define an alternative model that has less power, and but is more suitable for discussing the two algorithms.

A Weaker Model – The Comparison Model

Let us describe the model on the dictionary search problem as a game between Alice and Bob.

Alice places $n \geq 2$ boxes — from left to right labeled as 1, 2, ..., n — in front of Bob, and tells Bob:

- Each box contains an integer.
- The integer in box i is less than that in box j, for any $1 \leq i < j \leq n$.

Let S be the set of integers in the boxes.

There is another box — labeled as q — that contains an integer.

A Weaker Model – The Comparison Model

Bob's mission is to determine whether $q \in S$.

The game goes in rounds, where in each round:

- Bob picks a box id i.
- Alice peeks into the box i and box q, and tells Bob (i) whether the two boxes contain the same integer, and (ii) if not, also which box has a larger integer. Note that Alice reveals only the comparison result, but not anything else (in particular, not the integers in the two boxes).

Bob's goal is to figure out whether $q \in S$ in the smallest number of rounds.

A Weaker Model – The Comparison Model

Binary search solves the problem in at most $\lceil \log_2(n+1) \rceil$ rounds — this gives us an upper bound on how many rounds are needed to solve the problem.

What is interesting are lower bounds. Here is the question at the core: **what is the smallest number of rounds needed to solve the problem in the worst case?**

Next we will show that any algorithm must perform at least $\lceil \log_2(n+1) \rceil$ rounds. In other words, binary search is already optimal.

A Weaker Model – The Comparison Model

Our argument is known as an adversary argument. The idea is to consider an Alice that tries her best to "sabotage", by doing everything possible in her capacity to increase the number of rounds.

Before the game starts, Alice fixes the integers in the n boxes to be 1, 3, 5, ..., $2n - 1$, respectively. She, on the other hand, does not fix immediately the integer in the box q. Instead, she initializes a set $Q = \{0, 2, 4, ..., 2n\}$. Note that $|Q| = n + 1$. The set Q includes all the possible values of q that Alice can choose.

A Weaker Model – The Comparison Model

Consider now the interaction in a round. Let i be the id of the box selected by Bob. The set Q can now be divided into two subsets:

- $Q_{>} = \{x \in Q \mid x > i\}$
- $Q_{<} = \{x \in Q \mid x < i\}$

Clearly, either $|Q_{>}| \geq |Q|/2$ or $|Q_{<}| \geq |Q|/2$. Alice acts as follows:

- If $|Q_{>}| \geq |Q|/2$, she tells Bob "box $q >$ box i", and then set Q to $Q_{>}$.
- Otherwise, she tells Bob "box $q <$ box i", and then set Q to $Q_{<}$.

The above strategy ensures that, after each round, Bob cannot rule out the possibility that q is one of the integers in Q.

A Weaker Model – The Comparison Model

Questions for you to think:

- (Easy) convince yourself that $\lceil \log_2(n+1) \rceil$ rounds are needed to solve the problem in the worst case, regardless of Bob's strategy.
- (Not so easy) convince yourself that, on inputs where $q \notin S$, any algorithm can be described as a binary tree of $n+1$ leaves.