Lower Bounds

2017년 11월 27일 월요일

오후 3:30

ppt 12-11th. 21:55

Lower Bound for Algorithms

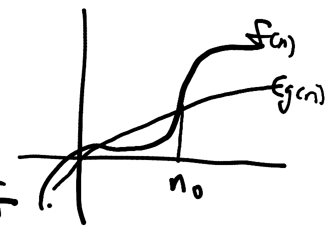
* review

Big-Omega :

$f(n)$ is $\Omega(g(n))$ if

$c\,g(n) \le f(n)$ for constant

c and $n > n_0$



⇒

* Let $T(A, f)$ be the # of functional evaluations performed by some algo A given the step function f as input

- "Time complexity of Algo A" in the worst-case running time over all the possible inputs

$$T(A) = \max_{|f|=n} T(A, f)$$

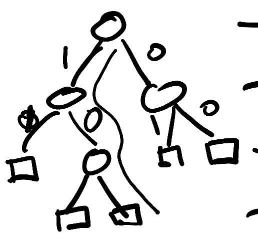
- "Time complexity of the problem" is time complexity of the <u>fastest</u> algo that correctly solves it.

$$T(n) = \min_{A} T(A) = \min_{A} \max_{|f|=n} T(A,f)$$

1. Comparison based Algo

Algo.s can be represented as Decision Trees

: rooted, ordered, binary.



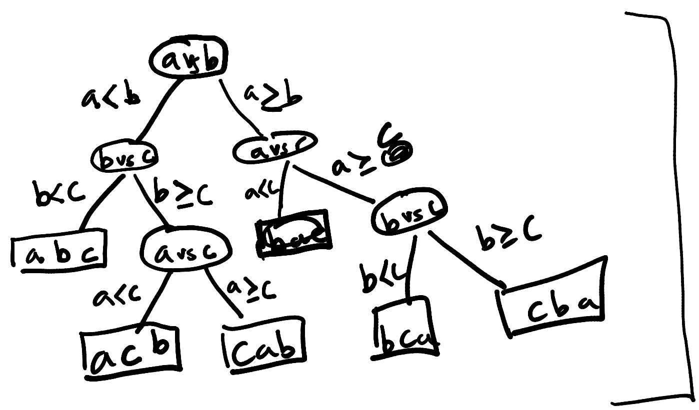
internal node = step function

leafs : output algo.

execution of algo gives path from root to a leaf (deterministic)

ex) d.t. for sorting $n=3$

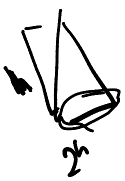
input $= \{a, b, c\}$



Theorem

Any decision tree that sorts n distinct elements has height at least $\log_2(n!)$.

- proof
  - Since the answer can be any of $n!$ permutation of the input of size n
    there are at least $n!$ leaves
  - Since there are at most 2 children of any non-leaf node
    a tree of height h can have at most 2^h leaves



$\therefore$ Thus the height must be at least $\lg_2 n!$

- corollary
  Any algo. that sort n elements by comparison req. $\Omega(n \lg n)$ time for some input of size n

proof) Any comparison based algorithm can be modeled by decision tree
There is going to be some leaf at least $\lg_2(n!)$.
Thus there is a computation that will perform at least $\underline{\lg_2(n!)}$ of comparisons.

$n/2$
$n/2$ $n/2$

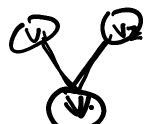
$$\log\left(n! \ge \underbrace{n(n-1)\cdots(n-n/2)}_{n/2} \ge \left(\tfrac{n}{2}\right)^{n/2} \right)$$

$$\lg_2 n! \ge \lg\left(\tfrac{n}{2}\right)^{n/2} = \frac{n}{2}\lg \frac{n}{2}$$

$$= O(n\lg n)$$

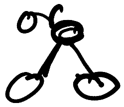
$$\therefore \Omega(n\lg n)$$

- Algebraic Computational Tree (ACT)
  - input value $V = \{x_1, \ldots, x_n\}$
  - non-leaf node v :
    are either with 1) one child which contains instructions



- $f_v \leftarrow f_{v_1}$ op f_{v_2} $\quad op \in \{+, -, \times, \div\}$

- $f_v \leftarrow c$
- $f_v \leftarrow \sqrt{f_{v_i}}$
- $f_v \leftarrow x_i$



2) with 2 chidren which contains
- $f_v > 0 \quad f_v < 0 \quad f_{v_1} = 0$ (yes or no)

- leaf nodes: labeled with binary strings

If it is Y/N answer, then 0 or 1

if it is combinatorial answer:

binary string that encodes the information output

claim

sorting takes $\Omega(n \lg n)$ time in ACT model

proof)

there must be at least $n!$ leaves for all of the input permutation.

The # of children of the node is $\boxed{\text{at most } 2}$. Thus the height of the tree must be at least $\lg(n!)$, as before in decision tree.

* Problem Reduction

Given 2 problem A and B, A is $f(n)$-tranformable to B if

1) any instance of A can be tranformed into an instance of B in $O(f(n))$ time

and 2) the solution for B can be transformed into a solution of A in $O(f(n))$ time where n is the size of the problem

* Let A be $f(n)$ - transformable to B

1) If A requires $g(n)$ time, the B takes at least $g(n) - cf(n)$ time

$$\begin{array}{ccc} A & \xrightarrow{f(n)\text{-tran}} & B \\ g(n) & \cdots\cdots\to & \boxed{g(n) - c\,f(n)} \end{array}$$

2) If B is solvable in $h(n)$ time then A is solvable in

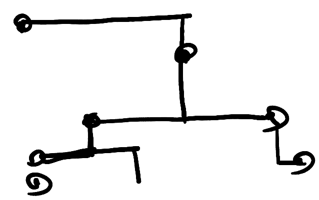
$c'f(n) + h(n')$ time, for

some const. $c' > 0$ and n' the size of tranfromed input.

$$\begin{array}{ccc} A & \xrightarrow{f(n)\text{-tran}} & B \\ \boxed{c'f(n) + h(n')} & \leftarrow\cdots\cdots & h(n') \end{array}$$

Problem : Rectilinear Steiner tree

Given a set of poin in the plane, find a set of horizantal and vertical line segments of miminal total length that connects all the points



• $O(n)$ time reduction from sorting (A) to RST (B)

$O(n)$ 1. sorting input : $\{v_1 \dots v_n\}$
transform input
$p_i = (v_i, 0)$ for $i = 1 \dots n$

⇒ Given the solution to RST

$O(n)$ Search the solution starting from the leftmost end point and print of v_i when each point p_i is encountered

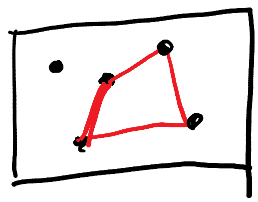
$g(n) - c f(n)$

$\Omega(n \lg n) - \boxed{\cancel{O(n)}}$

RST $\Omega(n \lg n)$

Problem 2: Maximum Empty Convex Region Problem

Given a rectangle in the plane and n points inside the rectangle, Find the largest convex region on the rectangle that contains no points and output the boundary of the region in counter clockwise order



Lower Bound using sorting

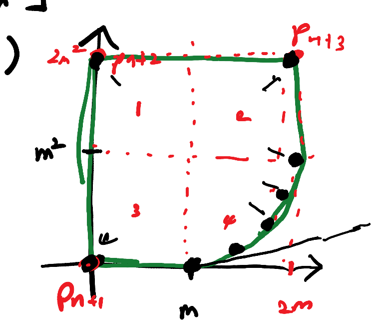
Reduce from sorting n positive value $\{v_1 \cdots v_n\}$ to MECR

① Input transform.

Let $m = \max\{v_1 \ldots v_n\}$

② $p_i = (\underbrace{m + v_i}_{x}, \underbrace{v_i^2}_{y})$

$\Theta(n)$ $\begin{cases} p_{n+1} = (0, 0) \\ p_{n+2} = (0, 2m^2) \\ p_{n+3} = (2m, 2m^2) \end{cases}$



② Output transformation

$\Theta(n)$ to translate solution of MECR

Start at (0,0) and walk around the region output the corresponding v_i values.

Since reduction takes $\Theta(n)$ time but we have $\Omega(n \lg n)$ for sorting the lower bound of MERR

$$g(n) - cf(n) = \underbrace{\Omega(n\lg n)} - c\Theta(n)$$

$$\therefore \Omega(n \lg n)$$

* other problems that has $\Omega(n\lg n)$ in ACT model

① element uniqness problem

- given n ~~[illegible]~~ numbers, determine if the elements are distincts.

2) set equality on inclusion

→ given set $A = \{x_1, \dots x_n\}$ and set $B = \{y_1, \dots y_n\}$, determine if $A = B$ or $A \subseteq B$

3) set disjointness P.

→ determine if $A \cap B = \phi$