Analysis and Design of Algorithms

NUS
National University of Singapore

Algorithms
CS3230

Tutorial

Week 5

Question 1

There exists a comparison sort of 5 numbers that uses at most 6 comparisons in the worst case.

- True
- False

Answer

Answer: False

Number of leaves in a decision tree which sorts 5 numbers: 5! = 120.

With 6 comparisons, we have up to 2^6 possible outputs, which is fewer than 120, hence cannot produce 120 leaves.

Question 2

Consider an array A of n elements. Which of following statement is **incorrect** (in the comparison model)?

- We can make a heap from A in $O(n)$ time.
- We can traverse a tree binary search tree built from A in $O(n)$ time.
- We can make a balanced binary search tree T from A in $O(n)$ time.
- We can find a k-th smallest element of A in sub-quadratic time

Answer

Answer: C is false.

- Proof by contradiction: Suppose that C is true.
  - I.e. a balanced tree can be constructed in O(n) time.

- Consider the following algorithm for **sorting**:
  - Construct a balanced tree in O(n) time
  - Do an in order traversal of the tree, known to be possible in O(n) time.

- The above algorithm runs in O(n) time, contradicting the comparison sort lower bound.

Answer

A is true. A heap can be constructed in linear time. Discuss offline if you do not know how.

B is true. Tree traversal can be done in linear time as it passes through each element a constant number of times.

D is true. Build a heap using O(n) time. Extract the first k minimums using O(k log n) time.

- A better solution: Use worst case linear time select algorithm (covered in next lecture).

Question 3

There is a list of records of n students, **already sorted** by students' name. The final exam score of a student is an integer between 0 and 100. We now want to sort this list, first by score, then by name. What is the fastest way to do the sorting?

Answer

- Run counting sort on the exam score column.

- Stability ensures that the names remain sorted within the same score.
- Runtime is $\Theta(n+k)$ where k=101.
  - k = 101 since the score is from 0 to 100.

Question 4

A set of n integers in the range $\{1, 2, \ldots, n\}$ can be sorted by RADIX-SORT in $O(n)$ time by running COUNTING-SORT on each bit of the binary representation.

1. True
2. False

Answer

Answer: False

To represent numbers in {1, …, n} we need $\Theta(\lg n)$ bits.

Each iteration of counting sort takes $\Theta(n)$ time.

If we run counting sort on each bit, we will run $\Theta(\lg n)$ iterations, giving a total runtime of $\Theta(n \lg n)$.

What is the right digit size to get $\Theta(n)$ runtime for radix sort in this case?

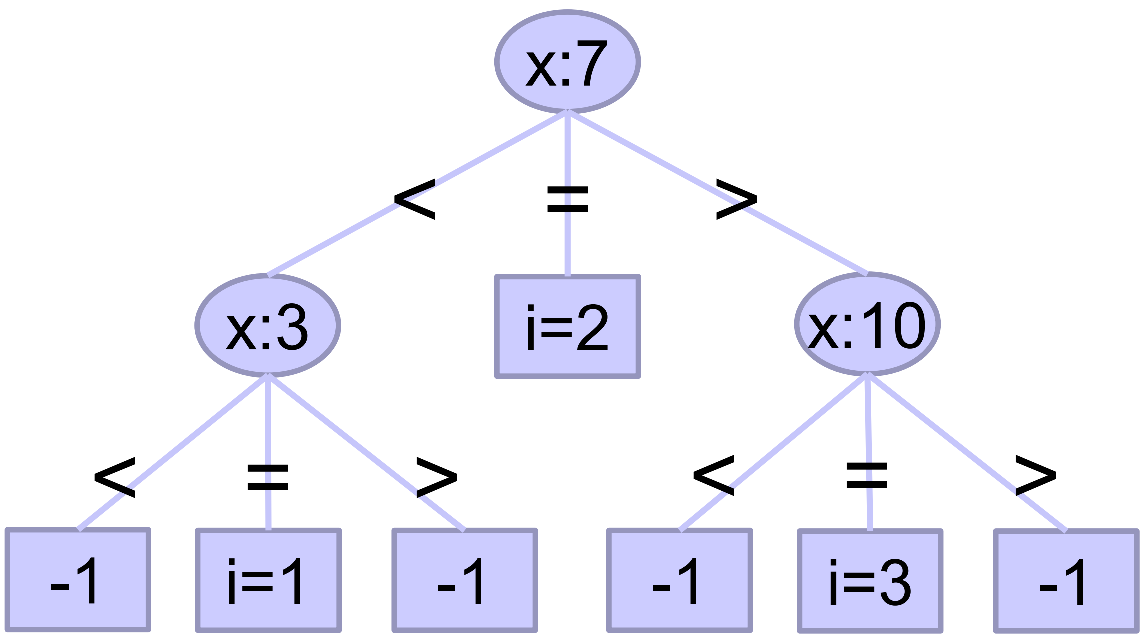
Question 5

- Given a sorted array of n real numbers A[1..n] and a query number x. You need to develop a function search(x, A) which returns an integer i if A[i]=x; and returns -1 otherwise.
- We have two assumptions:
  - Assume comparison model
  - Assume each comparison returns <, =, or >.
- What is the lower bound of the number of comparisons (exact bound instead of asymptotic bound)?

Answer

Example: the best decision tree for this problem for A=[3, 7, 10].

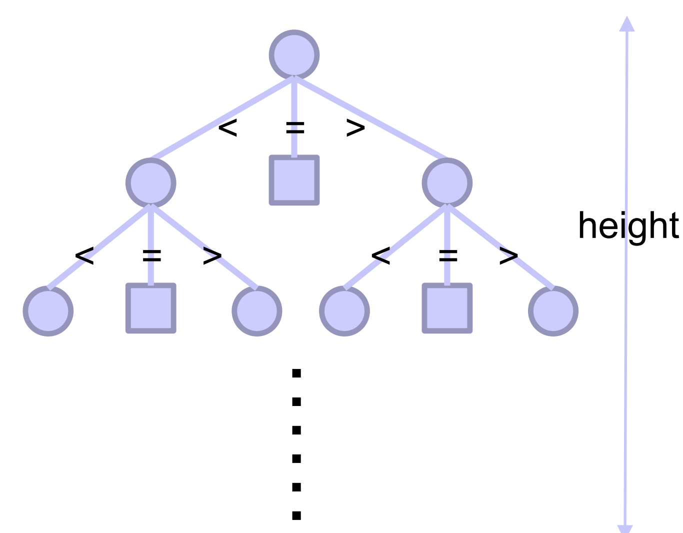
The lower bound is 2.



Answer

Answer: the lower bound is $\lfloor \lg n \rfloor + 1$.

- By contrary, assume there exists a decision tree with height ≤ lg n.
  - i.e. there exists an algorithm runs in lg n comparisons.
- For each answer i, where i=1,2,…,n, we need an internal node in depth smaller than or equal to lg n.
- There are $2^0 + 2^1 + \cdots + 2^{\lg n - 1} = 2^{\lg n} - 1 = n - 1$ internal nodes up to depth lg n.
- We arrived at contradiction.

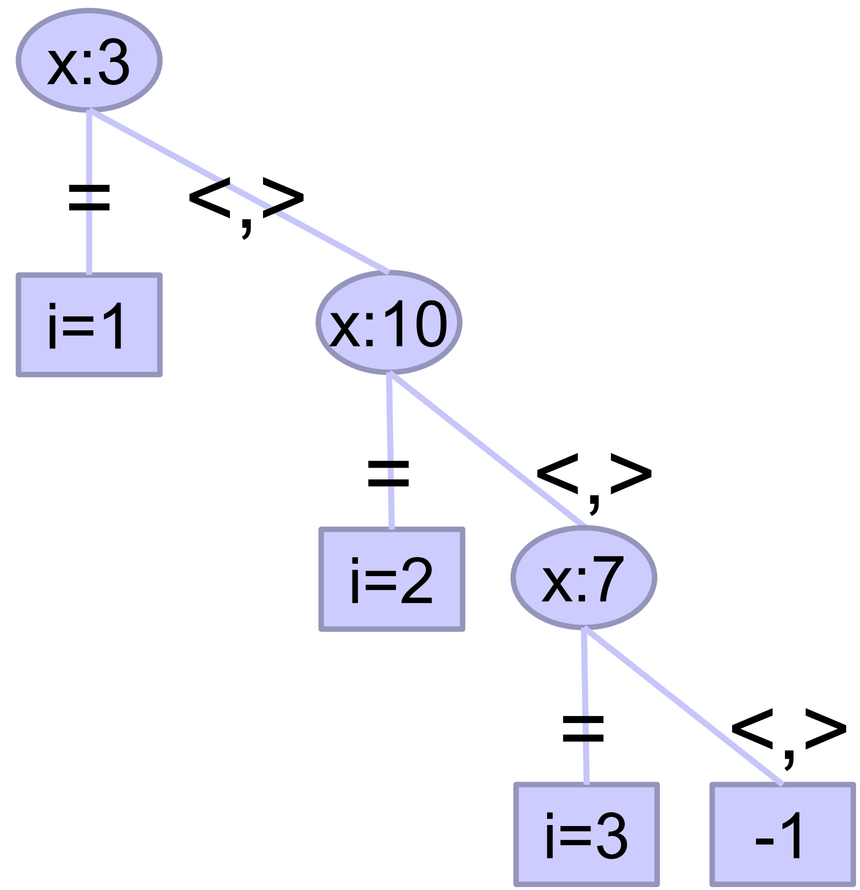


Question 6

- Given an unsorted array of n real numbers A[1..n] and a query number x. You need to develop a function search(x, A) which returns an integer i if A[i]=x; and returns -1 otherwise.
- We have two assumptions:
  - Assume comparison model
  - Assume each comparison returns <, =, or >.
- What is the lower bound of the number of comparisons (exact bound instead of asymptotic bound)?

Answer

Example: the best decision tree for this problem for A=[3, 10, 7].
The lower bound is 3.



Answer

Answer: the lower bound is n.

- By contrary, assume there exists a decision tree with height $\leq$ n-1.
  - i.e. there exists an algorithm runs in n comparisons.
- There exists a leaf for “-1”. The path from root to leaf contains at most n-1 comparisons.
- Each comparison tells the relationship between two numbers.
- Thus, n-1 comparisons can only tell us the relationship between x and at most n-1 numbers.
- Let A[i] be the number we don’t know its relationship with x.
- We can set A[i]=x.
- We arrived at contradiction.