Analysis and Design of Algorithms

NUS
National University of Singapore

Algorithms
CS3230

Tutorial

Week 7

Question 1

Given a set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ of points in the plane, what is best running time for finding the $\sqrt{n}$ points closest to the origin (0,0) (using normal Euclidean distance)?

- $\Theta(\sqrt{n})$
- $\Theta(n)$
- $\Theta(n \log n)$
- $\Theta(n\sqrt{n})$

Question 1

Answer: B $\Theta(n)$

Algorithm

- Compute distances to origin: $\Theta(n)$
- Run Select to find $\sqrt{n}$-th element: $\Theta(n)$
- Run Partition to get set of $\sqrt{n}$-th smallest elements: $\Theta(n)$
- Total: $\Theta(n)$

This is also the best possible as we need to at least look at every element: $\Omega(n)$ time.

Question 2

It is possible to modify quicksort to run in **worst case time** of $\Theta(n \lg n)$ by changing the pivot selection method.

- True
- False

Question 2

Answer: True

- Linear time select to find median: $\Theta(n)$
- Partition: $\Theta(n)$
- Total time for divide and combine: $\Theta(n)$
- Total: $T(n) = 2T(n/2)+\Theta(n) = \Theta(n \log n)$

Question 3

Consider the following three algorithms to obain the k smallest numbers in an array of n numbers in **sorted order**:

I: Sort the numbers. Then list out the first k numbers.

II: Build a min-heap. Then run Extract-Min k times to get the smallest k elements.

III: Use order-statistic algorithm to find the k^{th} smallest number. Partition around that number. Sort the k smallest numbers after partitioning.

Order the algorithms by slowest to fastest asymptotically in n and k, where $k \leq n$, assuming comparison-based algorithms are used.

1. I, II, III
2. III, II, I
3. II, I, III
4. I, III, II
5. II, III, I

Question 3

Answer: I, II, III

I. Sort then list out *k* numbers
   a) Sort: $\Theta(n \log n)$
   b) List out *k* numbers: $\Theta(k)$
   c) Total runtime: $\Theta(n \log n + k)$

II. Build min-heap. Extract-min k times.
   a) Build min-heap: $\Theta(n)$
   b) Extract min: $\Theta(k \log n)$
   c) Total runtime: $\Theta(n + k \log n)$

III. Run select to find k-th smallest number. Run partition. Sort the k smallest numbers.
   a) Select: $\Theta(n)$
   b) Partition: $\Theta(n)$
   c) Sort: $\Theta(k \log k)$
   d) Total runtime: $\Theta(n + k \log k)$

Question 4

Suppose we are given constants a and p where a is a positive integer and $0 < p < 1$. Assume the worst case linear time select algorithm is used to do pivot selection for quicksort such that:

- Case 1: the pivot will partition the array into subarrays of size pn and $(1 - p)n$ respectively.
- Case 2: the pivot will partition the array into subarrays of size a and $n - a$ respectively.

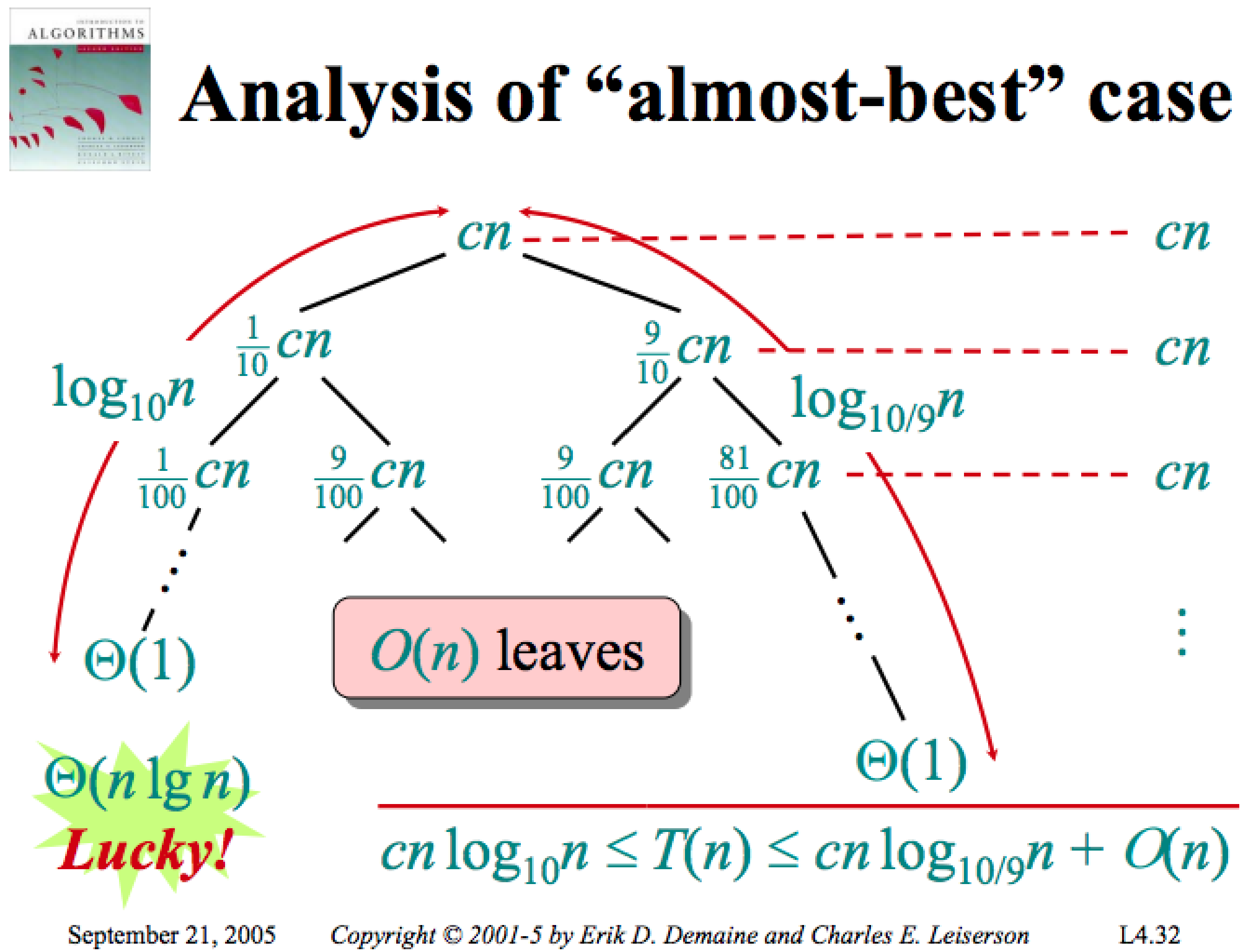
What are the runtimes for the two cases?

Question 4

Answer:

Case 1:

$\Theta(n \lg n)$

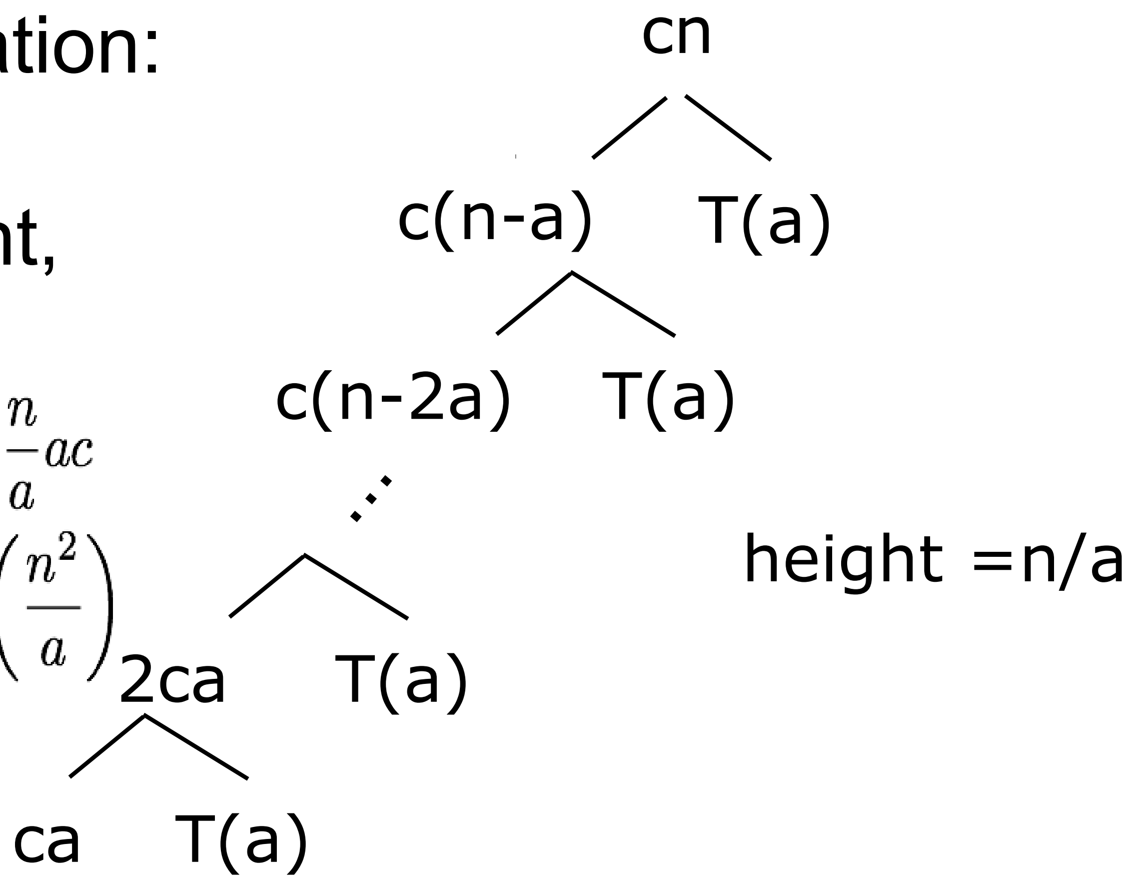


Question 4

Case 2: *T*(*n*) = *T*(*n*-*a*) + *T*(*a*) + *cn*
Recursion tree height is *n*/*a*
At depth *k*, computation:
T(a) + *c*(*n*-*ka*).
Summed over height,
we get

$$T(a)\frac{n}{a} + ac + 2ac + 3ac + \cdots + \frac{n}{a}ac$$

$$= T(a)\frac{n}{a} + \frac{ac}{2}\frac{n}{a}\left(\frac{n}{a}+1\right) \le C\left(\frac{n^2}{a}\right)$$



Question 5

Which algorithm should you select for finding the median in practice, if you are minimizing the expected runtime, and know that the input is selected by an adversary?

- Randomized quickselect
- Worst-case linear time Select

Solution

Answer: A

- The adversary cannot affect the expected runtime of randomized select.
- Both algorithms have linear expected runtime.
- So, it comes down to the constant multiplier for the linear term. Randomized select has a much better constant and is almost always preferred in practice.

Question 6

- Can you show that finding the median of 5 numbers requires at least 5 comparisons?
- Is it a tight bound? Can you give the best algorithm that find the median?

Solution

- For 5 numbers,
  - there are 5 choices of the median.
  - There are $\binom{4}{2}$ choices of the largest two numbers within the remaining 4 numbers.
- In total, there are $5\binom{4}{2} = 30$ combinations.
- Hence, the lower bound is $\lceil \lg 30 \rceil = 5$.

Solution

- This lower bound is not tight.
- The correct lower bound is 6
- The proof requires adversary argument (we will not discuss).

Solution

- There exists an algorithm that uses 6 comparison to find the median of 5 numbers.
- Let a_1, a_2, a_3, a_4, a_5 be the 5 numbers.
- 1. If $a_1 < a_2$, swap a_1 and a_2.
- 2. If $a_3 < a_4$, swap a_3 and a_4.
- 3. If $a_1 < a_3$, swap a_1 and a_3, a_2 and a_4.
- 4. If $a_2 < a_5$, swap a_2 and a_5.

$a_1 > a_2 \quad a_3 > a_4 \quad a_5$

Solution

- 5. If $a_2 > a_3$, step 6 else step 6'.
- 6. If $a_3 > a_5$, report a_3 else report a_5.
- 6'. If $a_2 > a_4$, report a_2 else report a_4.

$$a_1 > a_2 \quad a_3 > a_4 \quad a_5$$