PROBLEM SOLVING 3

Recursion, Recurrences, and Quicksort

CS340

Practice with recursion and recurrences

```
RM1(n)
  1 if n = 0
  2    return 0
  3 else
  4    return 1 + RM1(n - 1)
```

$$T(n) = \begin{cases} \Theta(1) & if\ n = 0 \\ T(n-1) + \Theta(1) & if\ n > 1 \end{cases}$$

What does this function do?
What is its theta?

Practice with recursion and recurrences

```
RM3(n)
  1 if n = 0
  2    return 0
  3 else
  4    return n + RM3(n - 1)
```

$$T(n) = \begin{cases} \Theta(1) & if\ n = 0 \\ T(n-1) + \Theta(1) & if\ n > 1 \end{cases}$$

What does this function do?
What is its theta?
Is there a Θ(1) way to do this?

Practice with recursion and recurrences

```
factorial(n)
  1 if n = 0
  2    return 1
  3 else
  4    return n * factorial(n - 1)
```

$$T(n) = \begin{cases} \Theta(1) & if\ n = 0 \\ T(n-1) + \Theta(1) & if\ n > 1 \end{cases}$$

What does this function do?
What is its theta?

Practice with recursion and recurrences

```
RM4(a, n)
  1 if n = 0
  2    return 1
  3 else
  4    return a * RM4(a, n - 1)
```

$$T(n) = \begin{cases} \Theta(1) & if\ n = 0 \\ T(n-1) + \Theta(1) & if\ n > 1 \end{cases}$$

What does this function do?
What is its theta?

Practice with recursion and recurrences

```
RM6(a, n)
  1 if n = 0
  2    return 1
  3 else if n mod 2 = 0
  4    return (RM6(a, n/2))^2
  5 else
  6    return a*(RM6(a, n/2))^2
```

$$T(n) = \begin{cases} \Theta(1) & if\ n = 0 \\ T(n/2) + \Theta(1) & if\ n > 1 \end{cases}$$

What does this function do?
What is its theta?

Recursion

- Can you write recursive code for Largest(int A[], int n) ?

Recursion vs iteration

- Can you write an iterative version of mergesort?

```
MERGE-SORT(A, p, r)
1  if p < r
2      q = ⌊(p + r)/2⌋
3      MERGE-SORT(A, p, q)
4      MERGE-SORT(A, q + 1, r)
5      MERGE(A, p, q, r)
```

Common Recurrences

- $T(n) = T(n/2) + \Theta(1)$ binary search $\Theta(\log n)$
- $T(n) = T(n-1) + \Theta(1)$ linear search $\Theta(n)$
- $T(n) = 2T(n/2) + \Theta(1)$ tree traversal $\Theta(n)$
- $T(n) = 2T(n/2) + \Theta(n)$ merge sort $\Theta(n \log n)$
- $T(n) = T(n-1) + \Theta(n)$ selection sort $\Theta(n^2)$

Best Sorting Method?

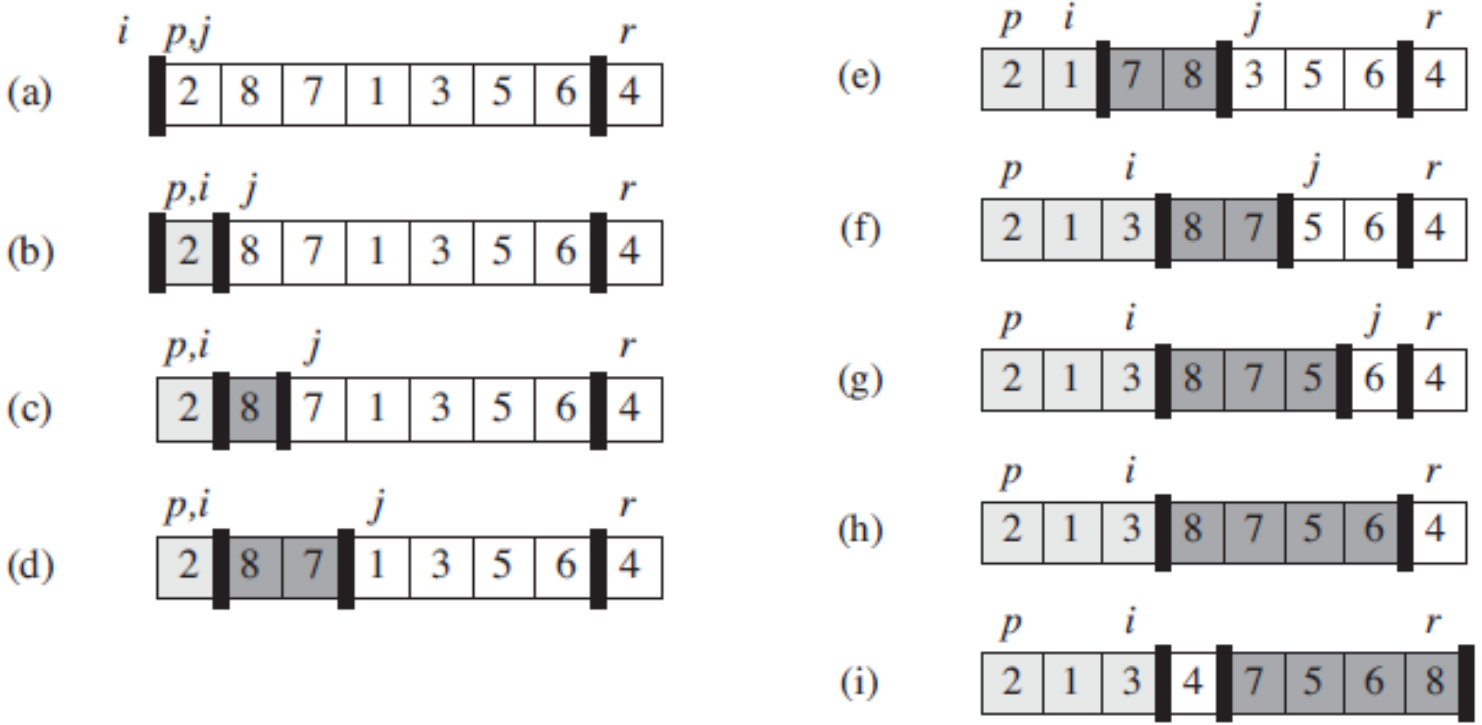
- Banks often record transactions on an account in order of the times of the transactions, but many people like to receive their bank statements with checks listed in order by check number. People usually write checks in order by check number, and merchants usually cash them with reasonable dispatch. The problem of converting time-of-transaction ordering to check-number ordering is therefore the problem of sorting almost-sorted input.
- Would INSERTION-SORT or QUICKSORT tend to do better on this problem?

Finding a Median…

- The median of an array of numbers is the number where half the remaining numbers are larger and half the remaining numbers are smaller.

- What is an intuitive algorithm for finding the median number in an array, and what is its time complexity?
- Can a median be found in O(n) time?
  - If so, how?

Quicksort Partition Method



What is the time complexity of partitioning?

Where is the median?

- Suppose we partition an array of size 1251 and the pivot ends up in position 867. What is the entire set of positions in which the median might be found? Assume no elements are equal to each other.

Largest

- Can the largest number in an array be found using the quicksort partition() method?
- What is the exact expected time complexity?

Improving Quicksort

- Suppose we have a O(n) time algorithm that finds the median of an unsorted array.
- Now consider a QuickSort implementation where we first find median using the above algorithm, then use median as pivot.
- What will be the worst case time complexity of this modified QuickSort.

Improving Quicksort

- Suppose we guarantee that Quicksort doesn't get stuck with an already-sorted array by:
  - Choose a pivot and run partition()
  - If the pivot is in the upper or lower quadrant of the array, simply discard it and choose another pivot until one is closer to the middle.
  - Do the areas < pivot and >pivot have to be exactly equal to get Θ(n lg n time)?
  - What does the recursion tree look like?
  - What's the time complexity of this variation on Quicksort?

Partition

- What is the result of partition() on the array:
  - 7 11 14 6 9 4 3 12

Doubling the input?

- Suppose it takes 200 secs to sort 1000 items using Quicksort.
- How long will it take to sort 2000 items?

Mergesort

- How could you adjust Mergesort so that it could be used in situations where you are short of memory?

```
MERGE-SORT(A, p, r)
1  if p < r
2      q = ⌊(p + r)/2⌋
3      MERGE-SORT(A, p, q)
4      MERGE-SORT(A, q + 1, r)
5      MERGE(A, p, q, r)
```

Interview Questions

1. Show how to implement a first-in, first-out queue with a priority queue.
2. Show how to implement a stack with a priority queue.

Interview Questions

- Give an O(n lg k)-time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists.
- (Hint: Use a min-heap for k-way merging.)

Consider the extremes:

- What is the result if we merge (n) 1-element lists?
- What is the result if we merge (2) n/2 element lists?

Interview Questions

Although merge sort runs in Θ(n lg n) worst-case time and insertion sort runs in Θ(n^2) worst-case time, the constant factors in insertion sort can make it faster in practice for small problem sizes on many machines. Thus, it makes sense to coarsen the leaves of the recursion by using insertion sort within merge sort when subproblems become sufficiently small. Consider a modification to merge sort in which n/k sublists of length k are sorted using insertion sort and then merged using the standard merging mechanism, where k is a value to be determined.

- Show that insertion sort can sort the n/k sublists, each of length k, in Θ(nk) worst-case time.
- b. Show how to merge the sublists in Θ(n lg n/k) worst-case time.
- c. Given that the modified algorithm runs in Θ(nk + n lg(n/k)) worst-case time, what is the largest value of k as a function of n for which the modified algorithm has the same running time as standard merge sort, in terms of Θ-notation?
- d. How should we choose k in practice?

Interview Questions

Express $n^3/1000 - 100n^2 - 100n + 3$ in terms of Θ notation.