In-Class Exercise

Solve the following recurrence relation:

$$T(n) = 8T(n/2) + 3n$$

$$T(1) = 1$$

Answer

Solve the following recurrence relation:

$T(n) = 8T(n/2) + 3n$

$T(1) = 1$

Find $T(n/2)$: $8(n/2^2) + 3(n/2)$

Plug in: $T(n)$: $8[8(n/2^2) + 3(n/2)] + 3n$

Find $T(n/4)$: $8T(n/2^3) + 3(n/2^2)$

Plug in: $T(n)$: $8^2\,[8T(n/2^3) + 3(n/2^2)] + 8 * 3(n/2) + 3n$

$= 8^3T(n/2^3) + 8^2 * 3n/4 + 8 *3n/2 + 3n$

Answer Cont'd

$8^3T(n/2^3) + 8^2 * 3n/4 + 8 *3n/2 + 3n$

$= 8^3T(n/2^3) + 3n(8^2/4 + 8/2 + 1)$

$= 8^3T(n/2^3) + 3n(4^2 + 4^1 + 4^0)$

- Pattern is:

$8^kT(n/2^k) + 3n[4^{k-1} +...+ 4^0)$

We stop recursion at $T(1) = 1$, or $n/2^k = 1$, or $n = 2^k$

Take lg of both sides to remove exponent: lgn = klg2, or k = lgn. Substitute lgn back in to the equation:

$8^{\lg n}T(1) + 3n[4^{\lg n-1} +...+ 4^0]$

$= 8^{\lg n}T(1) + 3n[(4^{\lg n} - 1) / 3]$ (if confused about this step, check the slide in the lectures labeled important math!

$= 8^{\lg n} + n(4^{\lg n} - 1)$

$= n^{\lg 8} + n(n^{\lg 4} - 1)$

$= n^3 + n(n^2 - 1)$

$= 2n^3 - n = \Theta(n^3)$

In-Class Exercise

Given that the comparison that takes place in the *else* statement is the basic operation, analyze Binary Search by finding its recurrence relation and determining its best-case and worst-case order.

```
index BinarySearch(index low, index high)
    if (low > high)
        return 0; // key not found in array
    else
        index middle = ⌊(low + high) / 2⌋
        if (x == S[middle])
            return middle
        else if (x < S[middle])
            return BinarySearch(low, middle - 1)
        return BinarySearch(middle + 1, high)
```

Answer

Base case low > high $\quad w(1) = 1$

$$w(n) = w\left(\tfrac{n}{2}\right) + 1$$

Assume implemented in Assembly efficiently, only one check.

$$w\left(\tfrac{n}{2}\right) = w\left(\tfrac{n}{4}\right) + 1 \rightarrow w(n) = \left[w\left(\tfrac{n}{4}\right) + 1\right] + 1$$

$$w\left(\tfrac{n}{4}\right) = w\left(\tfrac{n}{8}\right) + 1$$

$$w(n) = \left[w\left(\tfrac{n}{8}\right) + 1\right] + 1 + 1$$

$$w\left(\tfrac{n}{2^3}\right) + 3$$

$$w(k) = w\left(\tfrac{n}{2^k}\right) + k$$

stop at case 1 $\quad n/2^k = 1$

$2^k = n \quad k \lg 2 = \lg n \quad \lg 2 = 1$ so $\lg n = k$

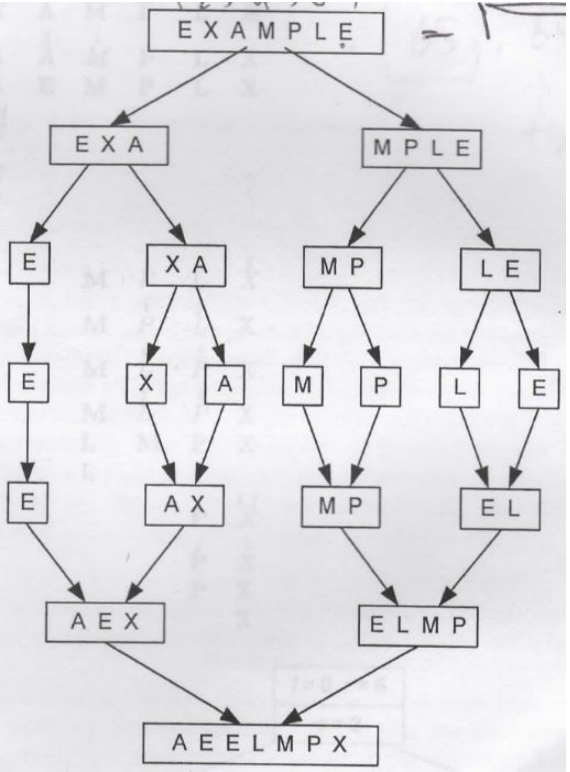
$1 + \lg n \quad \Theta(\lg n)$

In-Class Exercise

1. Sort [E_1, X, A, M, P, L, E_2] in alphabetical order using Mergesort. Show steps.
2. Sort 65, 60_1, 60_2, 60_3 in nondecreasing order using Mergesort. A sorting algorithm is called **stable** if it preserves the relative order of any two equal elements in its input. Is Mergesort stable? (hint: consider how the *Merge* procedure works)

Answers

1. Sort [E_1, X, A, M, P, L, E_2] in alphabetical order using Mergesort. Show steps.



Answers

2. Sort 65, 60_1, 60_2, 60_3 in nondecreasing order using Mergesort. A sorting algorithm is called **stable** if it preserves the relative order of any two equal elements in its input. Is Mergesort stable? (hint: consider how the *Merge* procedure works)

Mergesort is only stable if the comparison in merge is changed to <= rather than <

In-Class Exercise

S = {123, 34, 189, 56, 150, 12, 9, 240}

1. Show the recursion tree for sorting S with Quicksort.
2. Show the top-level partition on S.
3. Sort 65, 60_1, 60_2, 60_3 in nondecreasing order using Quicksort. A sorting algorithm is called **stable** if it preserves the relative order of any two equal elements in its input. Is Quicksort stable?

In-Class Exercise

S = {123, 34, 189, 56, 150, 12, 9, 240}

1. Show the top-level partition on S.

i	j	S[1]	2	3	4	5	6	7	8
-	-	123	34	189	56	150	12	9	240
2	2	123	34	189	56	150	12	9	240
3	2	123	34	189	56	150	12	9	240
4	3	123	34	189	56	150	12	9	240
5	3	123	34	56	189	150	12	9	240
6	4	123	34	56	189	150	12	9	240
7	5	123	34	56	12	150	189	9	240
8	5	123	34	56	12	9	189	150	240

Swap S[1] with S[j]

9 34 56 12 123 189 150 240