Date: Feb 24, 2016
Instructor: Professor Lawrence Teitelman

Analysis of Algorithms - CS 700/323
Lecture #4 – February 24, 2016

Notes by: Kok Teng Lee

Selection Sort

```
for i=0 to n-1
        find max and place it in position n-i-1
        swap(a,indexMax,n-1)
```

	Swap	Comparison
Best	n-1	n-1
Average	n-1	(n(n-1))/2
Worst	n-1	(n(n-1))/2

Insertion Sort

- Good for small array
- Data is mostly sorted

Sorted	UnSorted

Sentinel		25	36			30			

```
for i=1 to n-1
        temp = a[i]
        for j=i to 0
                if temp < a[j]
                        a[j+1] = a[j]
                else
                        a[j] = temp
                        break;
```

	Swap	Comparison
Best	n-1	n-1
Average	n-1	(n(n-1))/4
Worst	n-1	(n(n-1))/2

*Batch Insertion sort: Log n to search sorted array to find insertion point

**Sentinel is needed to ensure that it doesn't go out of bound

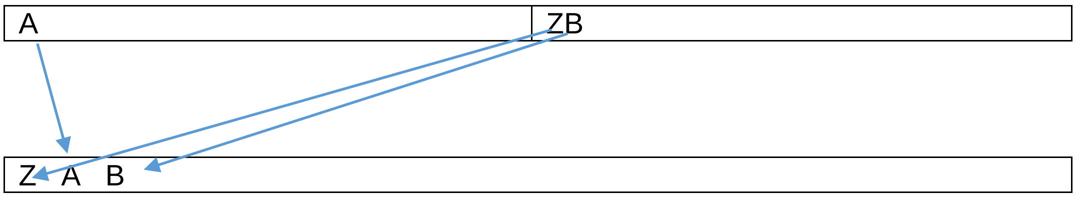
Date: Feb 24, 2016
Instructor: Professor Lawrence Teitelman

Shell Sort: $O(n^{1.5}) <= O(n^2)$

MergeSort

- Divide And Conquer

T(n) = 2T(n/2) + n – 1



Worst Case: n-1

(not in-place)

$T(n) = 2T(\frac{n}{2}) + n - 1,\ n = 2k$

$S(k) = T(n),\ S(0) = T(1) = 0$

$S(k\text{-}1) = T(\frac{n}{2})$

$$\frac{S(k)}{2^k} = \frac{2S(k-1)}{2^{k-1}} + \frac{2^{k-1}}{2^k}$$

Range Transformation:

$$R(k) = \frac{S(k)}{2^k}$$

$$R(k) = R(k-1) + 1 - \frac{1}{2^k}$$

$$R(k-1) - R(k-2) = 1 - \frac{1}{2^{k-1}}$$

$$R(1) - R(0) = 1 - \frac{1}{2^1}$$

$$k + 1 - [2 - (\tfrac{1}{2})^k] = k - 1 + (\tfrac{1}{2})^k$$

$$R(k) - 0 = k - 1 + (\frac{1}{2})^k$$

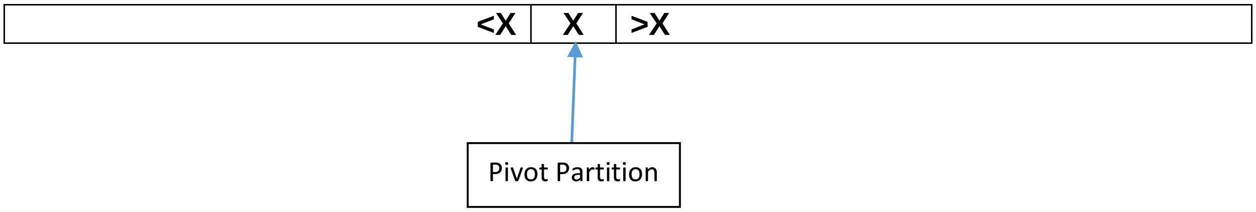
$$S(k) = 2^k, R(k) = 2^k[k - 1 + (\frac{1}{2})^k]$$

$$= 2^k.k - 2^k + 1 = 2^{lg\,n}.lg\,n - 2^{lg\,n} + 1 = nlg\,n - n + 1$$

Date: Feb 24, 2016
Instructor: Professor Lawrence Teitelman

Quicksort

In-place, Analysis in Section 14.2



Two-headed monster approach:

1. Left find bigger
2. Right find smaller
3. Swap (once both is found)

How to pick X:

1) First/Last
2) Random
3) Median of 3 scores

$T(n) = T(n-1) + n - 1$

Worst Case Like Bubblesort: $\frac{n(n-1)}{2}$

Best Case: n log n

Homework 3 Solutions:

Date: Feb 24, 2016
Instructor: Professor Lawrence Teitelman

Homework 3 let $f = 3n^3$

1) Prove that $3n^3 + 2n^2 + n = \Omega(1000n^2 + 2000n + 3000)$

$\cancel{\lim 3n^3}$ let $f(n) = 3n^3 + 2n^2 + n$

$g(n) = 1000n^2 + 2000n + 3000$

5/15

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{3n^3 + 2n^2 + n}{1000n^2 + 2000n + 3000}$$

$$\overset{L'H}{\Rightarrow} \lim_{n \to \infty} \frac{9n^2 + 4n + 1}{2000n + 2000} \overset{L'H}{\Rightarrow} \lim_{n \to \infty} \frac{18n + 4}{2000} = \infty$$

#

$$\therefore f(n) = \Omega(g(n))$$

2) $3n^3 + 2n^2 + n \geq \Omega(1000n^2 + 2000n + 3000)$

$$|f(x)| \geq C|g(x)|$$

$$3n^3 + 2n^2 + n \geq 1000(n^2 + 2n + 3)$$

$$C = \frac{1}{1000}$$

$$3n^3 + 2n^2 + n \geq C(n^2 + 2n + 3)$$

$\forall n \geq \underline{1}$

Date: Feb 24, 2016
Instructor: Professor Lawrence Teitelman

2.) Θ

$100\, n \log n + 50n = \Theta(50\, n \log n + \ldots$

let $f(n) = 100\, n \log n + 50n$

$g(n) = 50\, n \log n + 100\, n$

① $\rightarrow |f(x)| \geq C_1 |g(x)|$

② $\rightarrow |f(x)| \leq C_2 |g(x)|$

$C_1 |g(x)| \leq |f(x)| \leq C_2 |g(x)|$

① $100\, n \log n + 50n \geq 50\, n \log n + 100\, n$

$\cancel{2n \log n + n} \geq n \log n + 2n$

515

$C_1 = \frac{1}{50}$

$C_1 (50n \log n + 100n)$

$\leq C_2 (50\, n \log n + 100\, n)$

$C_2 = 2$

$C_2 (50n \log n + 100n)$

$C_1 (50\, n \log n + 100n) \leq 100\, n \log n + 50n$

$\leq C_2 (50\, n \log n + 100\, n)$

$C_1 = \frac{1}{50}$

$C_2 = 2$

$\forall\, n \geq 1$

3.)

Date: Feb 24, 2016
Instructor: Professor Lawrence Teitelman

3.) a.) if $f(n) = O(g(n))$ and $g(n) = O(f(n))$
✓ then $f(n) = g(n)$. False, it just mean
that $f(n) = \Theta(g(n))$

✓ b.) TRUE, it's transitive

✓ c.) ~~false~~ ~~it is theta.~~

$f(n) = O(g(n))$, $g(n) = O(h(n))$

$f(x) \leq g(n)$ | hypothesis: $h(n) = \Omega(f(n))$

$g(n) \leq h(n)$ | (TRUE)

✓ d.) if $f(n) = \omega(g(n))$, & $f(n) = o(h(n))$

$f(n) > g(n)$ & $f(n) < h(n)$

hypo: $h(n) = \Theta(g(n))$ False.

$g(n) < f(n) < h(n)$

✓ e.) $f_1(n) = O(g_1(n))$ & $f_2(n) = O(g_2(n))$,

$f_1(n) \times f_2(n) = O(g_1(n) \times g_2(n))$

$f_1(n) \leq g_1(n)$ & $f_2(n) \leq g_2(n)$

$\boxed{f_1(n) \times f_2(n) \leq g_1(n) \times g_2(n)}$

TRUE

Date: Feb 24, 2016
Instructor: Professor Lawrence Teitelman

4.) Best: $N-1$
AVG: $\frac{3}{2}N$
5/5
WORST: N

$2 \times (n-1)$

```
min = array[0]
max = array[0]
for (int i = 0; i < N; i++
  if (min > array[i])
    min = array[i];
  else if (max < array[i])
    max = array[i];
```

5.) Best: $N-1$
AVG: $\frac{n(n-1)}{2}$
WORST: $\frac{n(n-1)}{2}$

✓

$$(n-1) + \frac{n(n-1)}{2}$$

$$\frac{2(n-1) + n(n-1)}{2}$$

$$= \frac{(n-1)(2+n)}{2}$$

```
for (int i = length-1; i>
  boolean swap = false
  for (int j=0; j<i;
  {
    if (array[j] > array
    {
      swap = true
    }
  }
  if (!swap){break;}
```

Average $\frac{A_N^n}{n!}$

Date: Feb 24, 2016
Instructor: Professor Lawrence Teitelman

6.) public etc

```
int leftBound = 0;
int rightBound = myArray.length - 1
while (leftBound < rightBound ){
{
    boolean swap = false;
    for (int i = leftBound; i < rightBound
    {
      if (myArray[i] > myArray[i + 1]
       {  int temp = myArray[i];
          myArray[i] = myArray[i+1]
          myArray[i+1] = temp;
          swap = true
       }
    }
    rightBound--;
    for (int j = rightBound; j > leftBound
    {
       if(myArray[j] < myArray[j-1
       {  int temp = myArray[j];
          myArray[j] = myArray[j-1
          myArray[j-1]= temp;
          swap = true;
       }
    }
    leftBound++;
    if(!swap){ break;}
}
```

5/5 ✓

7.)

Date: Feb 24, 2016
Instructor: Professor Lawrence Teitelman

7.) $T(1)=0$; $T(n)=T(n-1)+n-1$

$T(2) = T(1) + 2 - 1$

$= T(1) + 1 = 0 + 1$;

$T(0) = 0$;

$T(1) = 0$;

$T(2) = 1$

Upper Bound $\leq O(n^2)$

Lower Bound $\geq O(n)$

$T(n) - \cancel{T(n-1)} = n-1$

$\cancel{T(n-1)} - \cancel{T(n-2)} = n-2$

$\cancel{T(2)} - T(1) = 1$

$T(n) - T(1) = \frac{(n-1)n}{2}$

$f(x) = ax^2 + bx + c$

$f(1) = 0 = a + b + \cancel{c}$

$f(0) = 0 = c$

$f(2) = 1 = 4a + 2b + \cancel{c}$

$a = -b$;

$4(-b) + 2b = 1$

$-2b = 1$

$b = -\frac{1}{2}$

$a = \frac{1}{2}$ #

$f(x) = \frac{1}{2}x^2 - \frac{1}{2}x$