**4.3-1**



**1. Substitution method**

If. T(n) < c\*n^2

T(n) < c\*(n-1)^2 + n

< c\* n^2 – 2cn + c + n

= c\*n^2 – ( 2c\*n - n –c) … desired–residual

<= c\*n^2.

Whenever c + n <= 2cn. For example, if c>=1, n>=1.

**2. recursion tree**

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

= T(n – 2) + (n – 1) + n

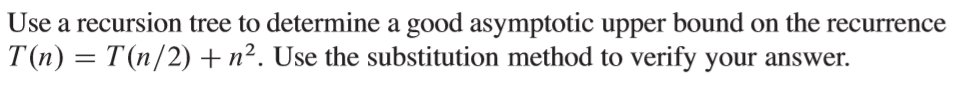
….

= T(0) + 1 + 2+ .. n-2 + n-1 + n ….. arithmetic series

= T(0) + n(n+1) / 2

= O(n^2)

**4.4-2**



n^2

(n/2)^2

(n/4)^2

….

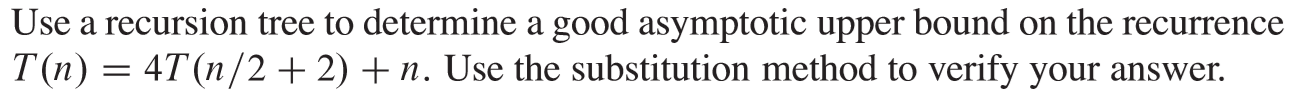
Therefore

= O()

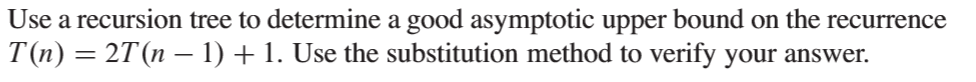
**Substiution**

Whenever = . T(n) = O()

**4.4-3**



**4.4-4**



**Recursion tree**

Tree is binary tree.

Height: n

Cost: 1

Max Num of Tree node: 2^h+1 – 1

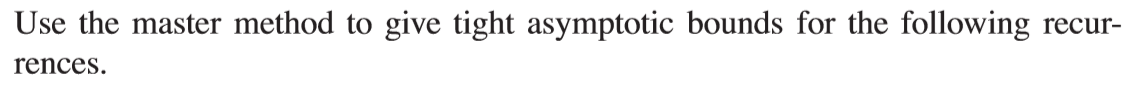
Therefore T(n) = O(2^n)

**Substiution**

– c

Whenever = . T(n) = O()

**4.5-1**





-> T(n) = Θ(



-> T(n) = Θ( lg n)



-> T(n) = Θ(



-> T(n) = Θ(

**7.1-1**



a. 13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11

b. 13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11

c. 13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11

d. 9, 19, 13, 5, 12, 8, 7, 4, 21, 2, 6, 11

e. 9, 5, 13, 19, 12, 8, 7, 4, 21, 2, 6, 11

f.. 9, 5, 13, 19, 12, 8, 7, 4, 21, 2, 6, 11

g. 9, 5, 8, 19, 12, 13, 7, 4, 21, 2, 6, 11

h. 9, 5, 8, 7, 12, 13, 19, 4, 21, 2, 6, 11

i.. 9, 5, 8, 7, 4, 13, 19, 12, 21, 2, 6, 11

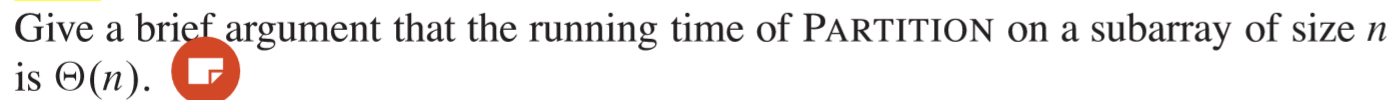
j.. 9, 5, 8, 7, 4, 13, 19, 12, 21, 2, 6, 11

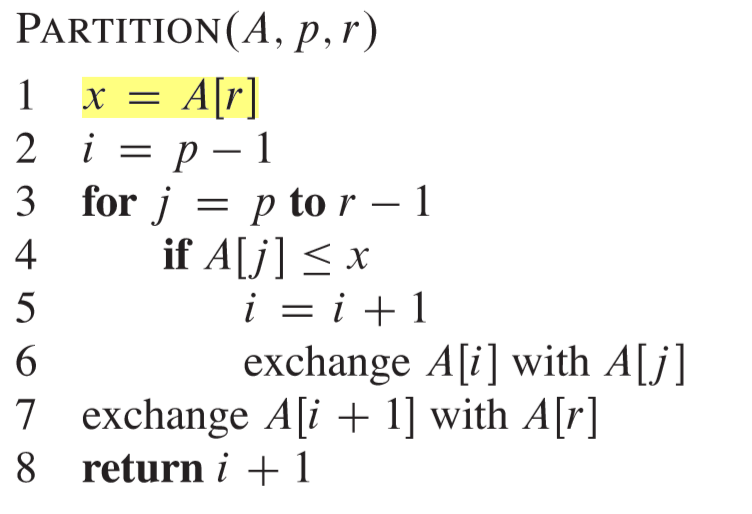
k. 9, 5, 8, 7, 4, 2, 19, 12, 21, 13, 6, 11

l. 9, 5, 8, 7, 4, 2, 6, 12, 21, 13, 19, 11

m. 9, 5, 8, 7, 4, 2, 6, 11, 12, 21, 13, 19

**7.1-3**



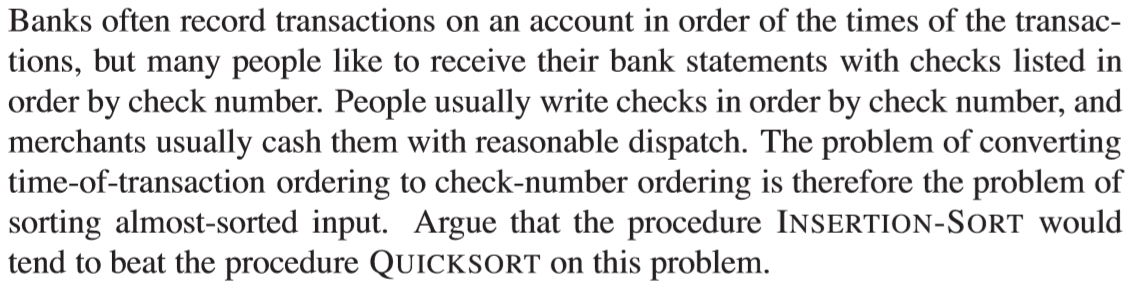


Subsrray of size = n = r – p. +1

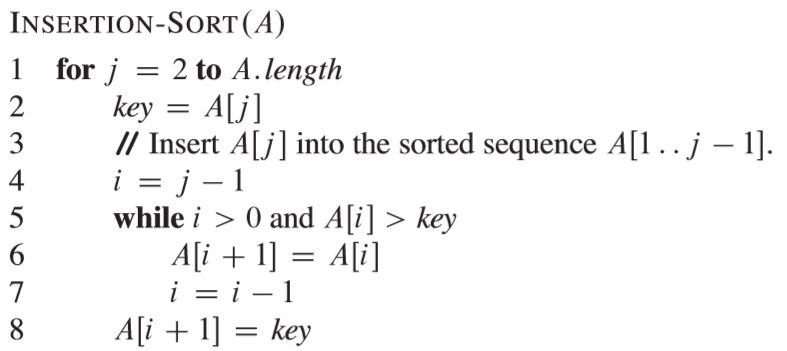
For loop iteration: r-p = n – 1

Therefore PARTITION = O(n)

**7.2-4**



Insertion sort:

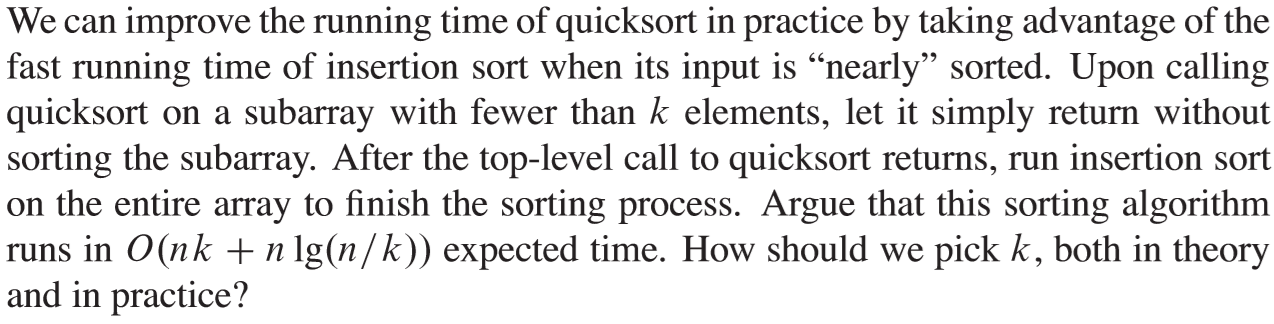


만일 배열 A가 작은 상수 c개를 제외한 모든 원소들이 잘 정렬 되어있고, 전체가 거의 정렬되어 있다면, while문은 c\*d 번 실행된다. (d는 아주 작은 상수) 따라서 이 문제는 Θ(n + c\*d), 즉 Θ(n)의 시간이 걸린다.

Quick sort:

배열 A가 전체적으로 거의 정렬되어있다면, partition은 n-1 : 1 로 나뉘어지는 경우의 수가 많아진다. 결국 worst case인 Θ(n^2) 이거나, expected case인 Θ(nlogn)룰 벗아나지 못한다.

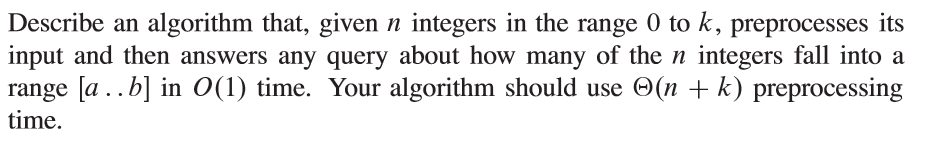
**7.4-5**

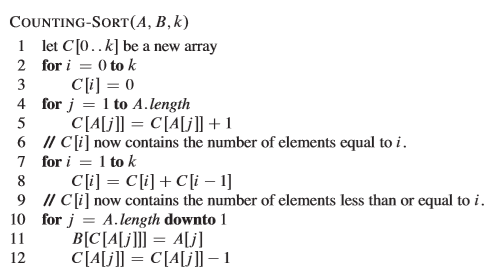


**8.2-2**

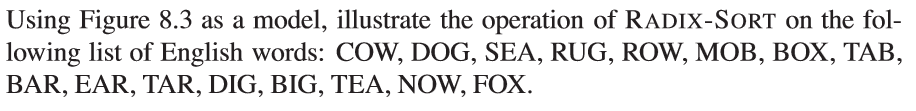


**8.2-4**

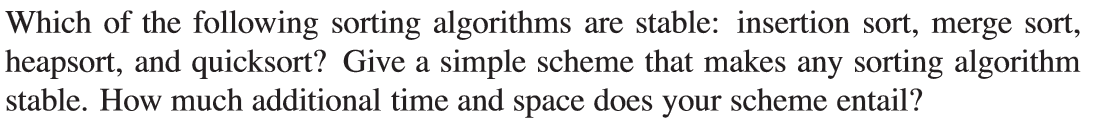




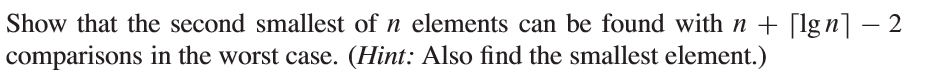
**8.3-1**



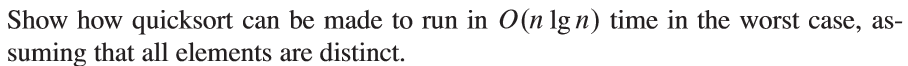
**8.3-2**



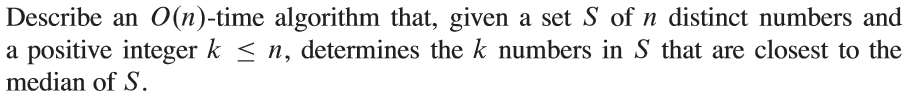
**9.1-1**



**9.3-3**



**9.3-7**



**9.3-9**