Homework 2

3) n — range of vals, m — size of value array

	Partition $(A, low, high)$	$T(n)$
C_n		
C_5	for(int $i = low + 1; i < high; i++$)	$n - 2$
C_6	if $A[low] == A[high]$	1
C_7	swap $A[i]$ and $A[high]$	1
C_8	break	
C_9	init	1
C_{10}	initial low, $l\_pivot = A[low]$	1
C_{11}	initial high, $h\_pivot = A[high]$	1
C_{12}		1
C_{13}	$l\_pivot = A[low]$	1
C_{14}	$h\_pivot = A[high]$	1
C_{15}	$low = low + 1$	1
C_{16}	$high = high + 1$	1
C_{17}	$current = low$	1
C_{18}	while $current <= high$ $(<= low)$	$n + (high - current$
C_{19}	if $A[current] <= l\_pivot$	1
C_{20}	swap $A[current]$ with $A[low]$	1.
C_{21}	$low += 1$	1
C_{22}	$current += 1$	1

3)

C_n		
	:	
C_{23}	if A [current] >= h_pivot (>=high)	1
C_{24}	swap A [high] with A[current]	1
C_{25}	high -= 1	1
C_{26}	else (> low, < high)	1
C_{27}	current += 1	1
	(end while)	
C_{28}	low = low - 1	1
C_{28}	high = high - 1	1
C_{31}	swap A [initial_low] with A[low]	1
C_{31}	swap A [initial_high] with A[high]	1
C_{32}	return (low, high)	1

(2)

		$T(n)$
C_A	Quicksort (Arr, low, high):	
C_p	if low < high:	1
$C.1$	p_low, p_high = partition (A, low, high)	f(n)
C_2	(<= low) Quicksort (A, low, p_low - 1)	
C_3	(> low, < high) Quicksort (A, p_low+1, p_high - 1)	
C_4	(>= high) Quicksort (A, p_high+1, high)	

$$T(p\_low - 1)$$
$$T(p\_high - p\_low)$$
$$T(n - p\_high + 1)$$

Homework 2

3) (3)

```
# assume every element has a duplicate
def dual_pivot_duplicate(arr):
    n = len(arr)
    for i = 1 to n
        for j = i+1 to n
            if arr[i] == arr[j]
                swap arr[j] w/ arr[i+1]
                break    (not used)
    call Quicksort(arr, low, high)
```

	T(n)
	1
c_{32}	n-1
c_{33}	n-2
c_{34}	1
c_{35}	1
c_{36}	1
c_{37}	
c_{38}	$\Theta(n)$

$$F(n) = (c_{32} + c_{34} + c_{35} + c_{36})(1) + c_{33}(n-1)$$
$$c_{34}(n-2) + c_{38} \cdot \Theta(n)$$

$$= \Theta(1) + \Theta(n)^2 + \Theta(n)$$
$$= \Theta(n^2) + \Theta(n),$$

$$T(n) = c_0(1) + c_1 F(n) + c_2 T(p\_low - 1)$$
$$+ c_3 T(p\_high - p\_low) + c_f T(n - p\_high + 1)$$

$$= \Theta(1) + \Theta(F(n)) + T(p\_low - 1)$$
$$+ T(p\_high - p\_low) + T(n - p\_high + 1);$$

Homework 2

3)

(4)

$$f(n) = (c_{10} \rightleftharpoons c_{17} + c_{19} \rightleftharpoons c_{32})(1) + c_5(n-2) + c_{18}(n)$$

$$= \Theta(1) + \Theta(n)$$
$$= \Theta(n),$$

(sub)

$$T(n) = \Theta(1) + \Theta(\Theta(n)) + T(g\_low - 1)$$
$$+ T(g\_high - g\_low) + T(n - g\_high + 1)$$

$$= \Theta(n) + T(g\_low - 1) + T(g\_high - g\_low)$$
$$+ T(n - g\_high + 1);$$

(sub)

$$T(n) = \Theta(n) + \Theta(\Theta(n) + T(g\_low - 1)$$
$$+ T(g\_high - g\_low) + T(n - g\_high + 1))$$

$$= \Theta(n) + T(g\_low - 1) + T(g\_high - g\_low)$$
$$+ T(n - g\_high + 1);$$

Best: $g\_high = n, g\_low = 1$ (all are same #)

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

$$= \Theta(n) + T(n-1) + 0$$

3) (5)

— Worst Case:

$$p1 \cong p2 \rightarrow p, \quad A[i] = \ldots \quad [i=n-1, j=n$$

$$T(n) = \text{maximize} \left(T(p-1) + T(p-p) + T(n-p) + \Theta(n) \right)$$

$$= \text{maximize} \left[T(p-1) + T(n-p) + \Theta(n) \right]$$

- Worst case partitioning:

$$T(n) = T(n-2) + T(0) + T(0) + \Theta(n^2)$$
$$= T(n-2) + \Theta(n)$$

$$T(n) = T(n-2) + cn$$

- $T(n-2)$, n must be $2k$ (even)

- $T(2k) = T(2k-2) + 2k, \quad T(2k) = T(2(k-1)) + 2k$

- $T'(k) = T'(k-1) + 2k = \dfrac{k(k+1)}{2} + (2k)^2$

- $T(2k) = 2 \cdot \dfrac{k(k+1)}{2} + 2k$

- $T(n) = \dfrac{n}{2}\left(\dfrac{n}{2} + 1\right) + n$

- $T(n) = \Theta(n^2);$

Homework 2

3) (6)
- Substitute $\Theta(n^2)$

- $T(n) = \text{maximize} \left[\Theta(p-1)^2 + \Theta(n-p)^2 \right] + \Theta(n)$
  $\leq c \cdot \max \left((p-1)^2 + (n-p)^2 \right) + \Theta(n)$

- $(p-1)^2 + (n-p)^2 \leq (n-2)$

- $f(p) = (p-1)^2 + (n-p)^2$

- $f'(p) = 2(p-1) - 2(n-p)$
  $= 2p - 2 - 2p + 2p$
  $= 4p - 2n - 2$

- $f'(p) = 4, \quad p = n-1;$

- $T(n) \geq \max \left(c, (q-1)^2 + c_1(n-n)^2 \right) + \Theta(n)$

- $T(an^2 + bn + c) \geq \max c, (an^2 + bn + c) + \Theta(n)$
  $= \Omega(n^2);$

$\rightarrow \Omega(n^2) \leq T(n) \leq O(n^2), \quad \boxed{T(n) = \Theta(n^2)};$

Homework 2

3) (7)

— Best Case: (unnecessary)

— $A[1] == A[2]$, all 3 partitions have same size
($partition $A[1:p-1]$ length $= A[p_2+1:h]$ length $= A[q_1,q_2]$
length.

• $T(n) = 3T\left(\dfrac{n-2}{3}\right) + \Theta(n) + \Theta(1 \cdot (2-1))$

$= 3T\left(\dfrac{n-2}{3}\right) + \Theta(n)$

$= 3T\left(\dfrac{n-2}{3}\right) + \Theta(n)$

$= 3T\left(\dfrac{n}{3}\right) + n$

L levels

$n \longrightarrow n$

$\dfrac{n}{3} \quad \dfrac{n}{3} \quad \dfrac{n}{3} \rightarrow n$

$\dfrac{n}{3^2} \quad \dfrac{n}{3^2} \quad \dfrac{n}{3^2} \rightarrow n$

$\left(\dfrac{n}{3}\right)^L \longrightarrow n$

Homework 2

3) (8)

$\frac{n}{3^L} = 1$, $n = 3^L$, $L = \log_3 n = \frac{\log n}{\log 3}$

$T(n) = \sum_{i=0}^{\log_3 n} n = \frac{n \log n}{\log 3}$,

$$\boxed{T(n) = \Theta(n \log n);}$$

Average case:

Each of