Morgan Baccus
CptS 350
Homework #3

Problem #1

```
x = A[q]
i = p-1
for j = p to q-1
If A[j] <= x
i = i+1
exchange A[i] with A[j]
exchange A[i+1] with A[q]
return i+1
```

Morgan Baccus

Problem #2

T_{avg1} → demanded average-case complexity

T_{avg} → average-case complexity

T_W → worst-case complexity

$\Theta(n^2)$ → complexity of intersort

$$T_{avg1}(n) = 1\% * T_W(n) + 99\% * T_{avg}(n)$$

Since $T_W(n) = \Theta(n^2)$ and $T_{avg}(n) = O(n^2)$, we see that...

$$\Theta(n^2) + 0 \le T_{avg1}(n) \le \Theta(n^2) + O(n^2) = \Theta(n^2)$$

Hence, $T_{avg1}(n) = \Theta(n^2)$

Morgan Baccus

Problem #3

$T(n) \rightarrow$ a run of iqsort over A with n elements

$\Theta(n) \rightarrow$ the cost of partition

$T_q(r-1) \rightarrow$ cost of quicksort on the low part

$T_i(n-r) \rightarrow$ cost of intersort on the high part

To make $T(n)$ be the best, we need to consider the cases when $T_q(r-1)$ and $T_i(n-r)$ are best.

Best quicksort case: $T_q(r-1) = \Theta[(r-1)\log(r-1)]$

Best intersort case: $T_i(n-r) = \Theta(n-r)$

We can see that the best case for quicksort cannot compete with the best case for intersort. So the best case for $T(n)$ is the best case of $T_i(n-r)$ when the high part is already sorted. When the low part is empty, the best case of $T(n)$ is $T_i(n-1)$ (best case) plus $\Theta(n)$. So the best case for iqsort is still $\Theta(n)$ which requires the input array to alredy be sorted.

To make $T(n)$ be the worst, we need to consider the cases when $T_q(r-1)$ and $T_i(n-r)$ are the worst.

Worst quicksort case: $T_q(r-1) = \Theta[(r-1)^2]$

Worst intersort case: $T_i(n-r) = \Theta[(n-r)^2]$

Since both of these are at the same level, the worst case for $T(n)$ can be achieved when either the low or high parts is empty. Hence, the worst case for iqsort is $\Theta[(n-1)^2] + \Theta(n) = \Theta(n^2)$. This requires either the array to only be decreasing or the array begins with a minimal element and is followed by a decreasing sub array.

Morgan Baccus

Problem #3 continued

Now we will return to this formula: $T(n) = \Theta(n) + T_q(r-1) + T_i(n-r)$

This formula only works for a fixed r.

We can use Tavg to denote the average-case complexity for iqSort:

$$T_{avg}(n) = \sum_{1 \le r \le n} \frac{1}{n}\left(\Theta(n) + T_{q\text{-}avg}(r-1) + T_{i\text{-}avg}(n-r)\right)$$

Using the original average-case quicksort and insert sort we get:

$$T_{avg}(n) \le \Theta(n) + \frac{1}{n} \sum_{1 \le r \le n} a * (r-1)\log(r-1) + b \times (n-r)^2$$

Using the above formula, we can show:

$$T_{avg}(n) \le \Theta(n) + \frac{1}{n} \sum_{1 \le r \le n} a * n\log n + b * n^2$$

$$T_{avg}(n) \le \Theta(n) + \frac{1}{n}\Theta(n^3)$$

$$\boxed{T_{avg}(n) = O(n^2)}$$

Morgan Baccus

Problem #4

$\Theta(n) \rightarrow$ the cost of partition

$T(r-1) \rightarrow$ the cost of mixsort over low part

$T_i(n-r) \rightarrow$ the cost of intersort over high part

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

Best case: To make $T(n)$ be the best, we need to consider the cases when $T(r-1)$ and $T_i(n-r)$ are the best.

Best case of $T(n)$: $T(n) = \Theta(n) + T(r-1) + T_i(n-r)$

Consider this formula:

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

$$T_W(n) \leq \Theta(n) + T_W(r-1) + T_{i\_w}(n-r)$$

$$\vdots$$

$$\boxed{T_W(n) \leq C * n^2}$$