Probabilistic Analysis
Solutions to Exercises

1. (a) In the worst case, c is always headsso steps 5-7 execute n^2 times and $T(n) \in \Theta(n^2)$.

   (b) Use the formula: $E(X) = \sum_{i=1}^{\infty} \text{Prob}(X \geq i)$.
   Let X = Number of heads.
   The running time is $cX + c$.
   The expected running time is $E(cX + c) = cE(X) + c$.

$$
\begin{aligned}
\text{Prob}(X \geq i) &= (1/2)^i \text{ for } i \leq n^2. \\
\text{Prob}(X \geq i) &= 0 \text{ for } i > n^2. \\
E(X) &= \sum_{i=1}^{\infty} \text{Prob}(X \geq i) = \sum_{i=1}^{n^2} \text{Prob}(X \geq i) + \sum_{i=n^2+1}^{\infty} \text{Prob}(X \geq i) \\
&= \sum_{i=1}^{n^2} \text{Prob}(X \geq i) + \sum_{i=n^2+1}^{\infty} 0 \\
&= \sum_{i=1}^{n^2} \text{Prob}(X \geq i) = \sum_{i=1}^{n^2} (1/2)^i = 1/2 + (1/2)^2 + (1/2)^3 + (1/2)^4 + \ldots + (1/2)^{(n^2)} \\
&\leq 1/2 + (1/2)^2 + (1/2)^3 + (1/2)^4 \ldots = (1/2)(1 + (1/2) + (1/2)^2 + (1/2)^3 + \ldots) \\
&= (1/2)(2) = 1. \\
ET(n) &= cE(X) + c \leq 2c. \\
ET(n) &= cE(X) + c \geq c.
\end{aligned}
$$

   Since $c \leq ET(n) \leq 2c$, expected running time $ET(n) \in \Theta(1)$.

2. Steps 3-9 take cn^2 time.

   (a) In the worst case, $k = n - 2$, so:

$$
\begin{aligned}
T(n) = cn^2 + T(n-2) &= \underbrace{cn^2 + c(n-2)^2 + c(n-4)^2 + \ldots + c}_{n/2} \\
&\leq \underbrace{cn^2 + cn^2 + \ldots + cn^2}_{n/2} = (n/2)cn^2 = (c/2)n^3. \\
T(n) &= \underbrace{cn^2 + c(n-2)^2 + c(n-4)^2 + \ldots + c}_{n/2} \\
&\geq \underbrace{cn^2 + c(n-2)^2 + c(n-4)^2 + \ldots + c(n/2)^2}_{n/4} \\
&\geq \underbrace{c(n/2)^2 + c(n/2)^2 + \ldots + c(n/2)^2}_{n/4} = (n/4)c(n/2)^2 = (c/4)n^3(1/4) = (c/16)n^3.
\end{aligned}
$$

   Since $(c/16)n^3 \leq T(n) \leq (c/2)n^2$, the worst case running time $T(n) \in \Theta(n^3)$.

(b) Let $ET(n)$ be the expected running time on an array of size n.
Note that

$$\begin{aligned}
ET(n|k < n/2) &\le ET(n|k = n/2) \le cn^2 + ET(n/2).\\
ET(n|k \ge n/2) &\le ET(n|k = n-2) \le cn^2 + ET(n-2).\\
ET(n) &= \text{Prob}(k < n/2)ET(n|k < n/2) + \text{Prob}(k \ge n/2)ET(n|k \ge n/2)\\
&\le (1/2)(cn^2 + ET(n/2)) + (1/2)(cn^2 + ET(n-2))\\
&= cn^2 + (1/2)ET(n/2) + (1/2)ET(n-2)\\
&\le cn^2 + (1/2)ET(n/2) + (1/2)ET(n).\\
(1/2)ET(n) &\le cn^2 + (1/2)ET(n/2).\\
ET(n) &\le 2cn^2 + ET(n/2).\\
ET(n) &\le c'n^2 + ET(n/2) \text{ for } c' = 2c\\
&\le c'n^2 + c'n^2(1/2)^2 + c'n^2((1/2)^2)^2 + c'n^2((1/2)^3)^2 + \ldots + c'\\
&= c'n^2(1 + (1/2)^2) + ((1/2)^2)^2) + ((1/2)^3)^2 + \ldots + (1/n^2))\\
&= c'n^2(1 + (1/2)^2 + ((1/2)^2)^2 + ((1/2)^2)^3 + \ldots + (1/n^2))\\
&\le c'n^2(1 + (1/4) + (1/4)^2 + (1/4)^3 + \ldots) = c'n^2\frac{1}{1-(1/4)} = (4/3)c'n^2.
\end{aligned}$$

The algorithm always takes cn^2 time for steps 3-10 no matter what value k has, so $ET(n) \ge cn^2$. Since $cn^2 \le ET(n) \le (4/3)c'n^2$, expected time $ET(n) \in \Theta(n^2)$.

3. Steps 3-6 take cn time.

(a) In the worst case, k equals 1.

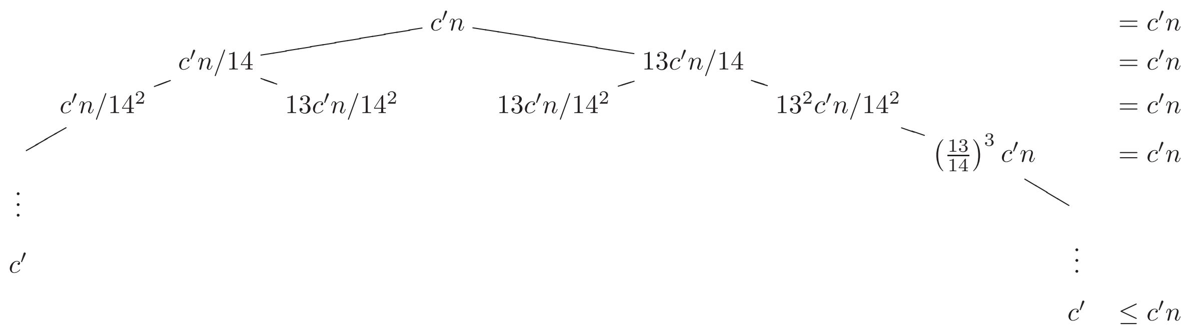
$$T(n) = cn + T(1) + T(n-1) = cn + c + T(n-1) \approx cn + T(n-1).$$
$$T(n) = cn + T(n-1) = cn + c(n-1) + c(n-2) + \ldots + c = cn(n+1)/2 \in \Theta(n^2).$$

(b) Let $ET(n)$ be the expected running time on an array of size n.

$$\begin{aligned}
ET(n|k < n/14) &\le ET(n|k = 1) \le cn + c + ET(n-1) \approx cn + ET(n-1).\\
ET(n|k > n/14) &\le ET(n|k = n/14) \le cn + ET(n/14) + ET(13n/14).
\end{aligned}$$

$$\begin{aligned}
ET(n) &= \text{Prob}(k < n/14)ET(n|k < n/14) + \text{Prob}(n/14 \le k)ET(n|n/14 \le k)\\
&= \frac{1}{2}ET(n|k < n/14) + \frac{1}{2}ET(n|k \ge n/14)\\
&\le \frac{1}{2}(ET(n-1) + cn) + \frac{1}{2}\Big(ET(n/14) + ET(13n/14) + cn\Big)\\
&= cn + \frac{1}{2}ET(n-1) + \frac{1}{2}\Big(ET(n/14) + ET(13n/14)\Big)\\
&\le cn + \frac{1}{2}ET(n) + \frac{1}{2}\Big(ET(n/14) + ET(13n/14)\Big).\\
\frac{1}{2}ET(n) &\le cn + \frac{1}{2}\Big(ET(n/14) + ET(13n/14)\Big).\\
ET(n) &\le 2cn + ET(n/14) + ET(13n/14).\\
ET(n) &\le c'n + ET(n/14) + ET(13n/14) \text{ for } c' = 2c.
\end{aligned}$$

Recursion tree:



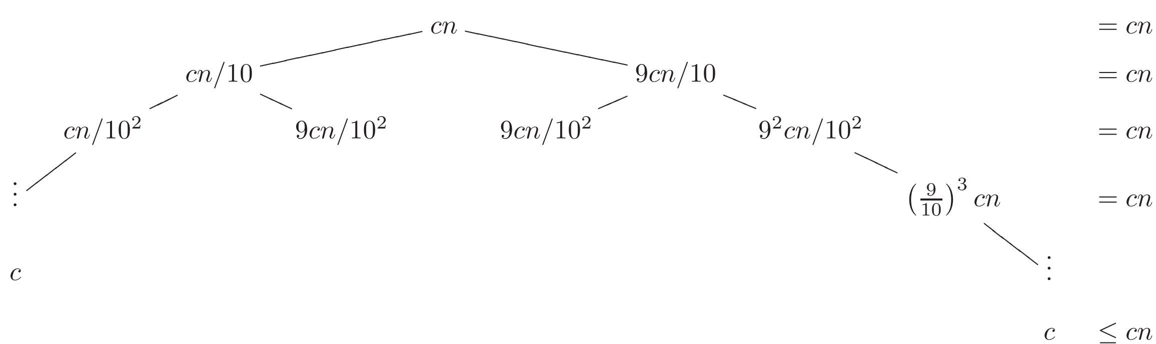
The height of this tree is $\log_{14/13}(n)$ so the total work at all nodes is at most $cn\log_{14/13}(n)$.
Thus, $ET(n) \leq cn\log_{14/13}(n) = c'n\log_2(n)$ and $ET(n) \in O(n\log_2(n))$.
In the best case, k equals $\lfloor n/7 \rfloor$ and $ET(n) = cn + ET(n/7) + ET(6n/7)$.
Using a recursion tree similar to the one used above, the shortest path to a leaf is $\log_7(n)$ and the total work is at least $cn\log_7(n) = c''n\log_2(n)$.
Thus, $c''n\log_2(n) \leq ET(n) \leq c'n\log_2(n)$, and expected running time $ET(n) \in \Theta(n\log_2(n))$.

4. Functions `NearCenter` and `Partition` take cn time.
Since `NearCenter` always returns an element whose rank is between $n/10$ and $9n/10$, the worst case is when `NearCenter` returns an element of rank $n/10$ (or rank $9n/10$.)

In the worst case where `NearCenter` returns an element of rank $n/10$, the running time is:

$$T(n) = cn + T(n/10) + T(9n/10).$$

Recursion tree:



The height of this tree is $\log_{10/9}(n)$ so the total work at all nodes is at most $cn\log_{10/9}(n)$.
Thus, $T(n) \leq cn\log_{10/9}(n) \in O(n\log_2(n))$.

In the best case, k equals $n/2$ so

$$\begin{aligned} T(n) &\geq cn + 2T(n/2) \geq cn + 2(cn/2 + 2T(n/2^2)) = cn + cn + 2^2T(n/2^2) \\ &= \underbrace{cn + cn + \ldots + cn + 2^{\log_2(n)}T(1)}_{\log_2(n)} = \underbrace{cn + cn + \ldots + cn + cn}_{\log_2(n)} = n\log_2(n) \in \Omega(n\log_2(n). \end{aligned}$$

Thus, $T(n) \in \Theta(n\log_2(n))$.