HW1 Solution

CS 498: Algorithms on big data, Spring 2019 Version: **1.0**

Submitted by:

- $\ll$**Ray Ying**$\gg$: $\ll$**xinruiy2**$\gg$
- $\ll$**Aditya Pillai**$\gg$: $\ll$**apillai4**$\gg$

1

1. We know from the last two sentences that X is the random Variable donating the output value and α is the true average. By Chebyshev's Inequality,

$$Pr[|X-\alpha| \geq \epsilon] \leq \frac{Var(X)}{\epsilon^2}$$

However, we want to show that

$$Pr[|X-\alpha| \geq \epsilon] \leq \delta$$

Therefore, we only need to show that

$$\frac{Var(X)}{\epsilon^2} \leq \delta$$

δ cannot be 0 since δ can be denominator, we can transform the above inequality to

$$\frac{Var(X)}{\delta\epsilon^2} \leq 1$$

As we are given that

$$\frac{(b-a)^2}{\delta\epsilon^2} \leq k$$

We want to prove that $\frac{(b-a)^2}{k} \geq Var(X)$, as X is the mean of the all X_i where X_i are the height of people in the k sample, $X = \sum_i \frac{X_i}{k}$.

Based on the definition of $Var(X)$, since X_i and X_j where $i \neq j$, they are independent Variables, then $Var(X) = \sum Var(X_i/k)$.

We can prove that $Var(X_i) \leq (b-a)^2/4$. Since $X_i \leq b$, $X_i \leq b \sum_i b \cdot X_i \geq \sum_i X_i^2$. So $Var(X_i) = E(X_i^2) - E(X_i)^2 \leq E(b \cdot X_i) - E(X_i)^2 = b \cdot E(X_i) - E(X_i)^2 = E(X_i)(b - E(X_i))$. Since $b - a \geq b - E(X_i)$, $Var(X_i) \leq E(X_i)(b-E(X_i)) \leq (b-a)^2/4$.

Therefore, $Var(X_i/k) \leq \frac{((b-a)/k)^2}{4}$ as $x_i/k \in [a/k, a/k]$, therefore, $Var(X) \leq k \cdot \frac{((b-a)/k)^2}{4} \implies$ Var(X) $\leq \frac{(b-a)^2}{4k} \implies$
k $\cdot Var(X) = (b-a)^2/4$

Hence, we have $(b-a)^2 \geq k \cdot Var(X) \implies \frac{Var(X)}{\delta\epsilon^2} \leq 1 \implies Pr[|X-\alpha| \geq \epsilon] \leq \delta$.

2. For Chernoff's inequality, we have the general form:

$$Pr[|X-\alpha| \geq \epsilon] \leq 2 \cdot e^{(\frac{-\epsilon^2}{2k})}$$

However, for Chernoff's inequality, we need to normalize each X to the range $[-1,1]$. Also, X should be the total sum, therefore, $X = kX$, $\alpha = k\alpha$ and $\epsilon = k\epsilon$. We have to normalize it to satisfy the precondition, we want to assume that X and α is in the range of $[-1,1]$, then X would be some constant $z + 2/(b-a)$ and α would be $z + 2/(b-a)$, however, as we are taking the absolute value of the difference, the constant doesn't matter. So our Chernoff's inequality will be:

$$Pr[|k\frac{2}{(b-a)} \cdot X - k\frac{2}{(b-a)}\alpha| \geq \frac{2}{(b-a)}k\epsilon] \leq 2 \cdot e^{(\frac{-(\frac{2}{(b-a)}k\epsilon)^2}{2k})}$$

We will perform transformation on the left side, the whole inequation will become

$$Pr[|X-\alpha| \geq \epsilon] \leq 2 \cdot e^{(\frac{-(\frac{2}{(b-a)}k\epsilon)^2}{2k})}$$

Therefore, all we need to show is

$$\delta \geq 2 \cdot e^{(\frac{-2k\epsilon^2}{(b-a)^2})}$$

From the given condition of $k \geq \frac{c(b-a)^2 log(2/\delta)}{\epsilon^2}$, we can do some transformations on the inequality. $k \geq \frac{c(b-a)^2 log(2/\delta)}{\epsilon^2} \implies (\text{k } \cdot \epsilon^2)/(c \cdot (b-a)^2) \geq log(2/\delta) \implies e^{(k \cdot \epsilon^2)/(c \cdot (b-a)^2)} \geq 2/\delta \implies \delta \geq 2/e^{(k \cdot \epsilon^2)/(c \cdot (b-a)^2)}$.

In order to prove that $\delta \geq 2 \cdot e^{(\frac{-2k\epsilon^2}{(b-a)^2})}$, we can prove that $2/e^{(k \cdot \epsilon^2)/(c \cdot (b-a)^2)} \geq 2 \cdot e^{(\frac{-2k\epsilon^2}{(b-a)^2})} \implies 1 \geq e^{(k \cdot \epsilon^2)/(c \cdot (b-a)^2)} \cdot e^{(\frac{-2k\epsilon^2}{(b-a)^2})} \implies e^{(k \cdot \epsilon^2)/(c \cdot (b-a)^2) + (\frac{-2k\epsilon^2}{(b-a)^2})} \leq 1$.

To prove that $e^{(k \cdot \epsilon^2)/(c \cdot (b-a)^2) + (\frac{-2k\epsilon^2}{(b-a)^2})} \leq 1$, we need to prove that $(k \cdot \epsilon^2)/(c \cdot (b-a)^2) - (\frac{2k\epsilon^2}{(b-a)^2}) \leq 0 \implies (\frac{2k\epsilon^2}{(b-a)^2}) \geq (k \cdot \epsilon^2)/(c \cdot (b-a)^2) \implies c \geq \frac{1}{2}$. Therefore, $c \geq \frac{1}{2}$, $\delta \geq 2/e^{(k \cdot \epsilon^2)/(c \cdot (b-a)^2)} \geq 2 \cdot e^{(\frac{-2k\epsilon^2}{(b-a)^2})}$. Hence, we showed that there exist a constant $c = \frac{1}{2} > 0$ that

$$Pr[|X-\alpha| \geq \epsilon] \leq \delta$$

2

1. The formal description of the algorithm is as following:
   (a) Choose one element uniform at random from the array and find its rank.
   (b) If the rank it's some where between $n/4$ to $3n/4$, then we choose this element as pivot. Otherwise, we repeat the previous step.
   (c) Perform quicksort on that pivot and divided the array into three subarray: those smaller than pivot, those larger than pivot, and the pivot itself.
   (d) Perform the small algorithm above on the smaller subarrays and concatenate the result.

   The Pseudocode for the algorithm is in the last page.
2. Let total running time of the randomized quicksort $T(n)$, and $E(T(n))$ be the expected running time of the randomized quicksort. For $T(n)$, the running time for each sort time will have two parts: choosing the pivot, sorting based on the good pivot. Since you will always end up with the good pivot and do the sorting. The expected running time for ranking good pivot is n(the list size) and the running time for sorting list based on this good pivot is n as well. Let z be the number of choosing bad pivots before getting the good pivot, the expected running time for

choosing the bad pivot is $\sum_z 1/2^z \cdot zn = n \cdot \sum_z 1/2^z \cdot z = 2n$. Therefore, the total expected running time will be $4n$. As for the recursive steps, the running time will be based on the rank of the pivot. Let i be the rank of the pivot ranging from $[n/4, 3n/4]$, the expected running time for the recursive steps will be: $\sum_i (1/(n/2)) \cdot (T(i-1) + T(n-i))$. Therefore, the total expected $E(T(n)) = 4n + \sum_i (2/(n)) \cdot (T(i-1) + T(n-i))$. With base case $T(n) = 0$, in each level of recursive tree, the sum is in $O(n)$. There will be at most $log_{3/4} n$ levels. Therefore, $E(T(n))$ is $O(nlogn)$.

3. Let T_i be the comparisons performed at level i of the recursion. Then the run-time of the algorithm is $\sum_{i=1}^{M} T_i$, M is the number of levels and $M \leq \log_{4/3} n$ since the pivot is always chosen in a "good" region. Let $T_{i,k}$ be the number of comparisons done for the kth subarray in the ith recursive call, and n_k be the size of the subarray k at level i
   $\mathbb{E}(T_i) = \sum_{k=1}^{2^i} \mathbb{E}(T_{i,k}) = \sum_{k=1}^{2^i} 2n_k = 2n$ because the sum of the size of subarrays at any level is n and in expectation it take 2 tries to pick a good pivot, each try costs n_k comparisons
   $\Pr(T_i > 8n/3) \leq \mathbb{E}(T_i)/(8n/3) = 3/4$. The number of comparisons it takes across different levels is independent so the probability it takes more than 8n/3 comparisons across all levels is $(\frac{3}{4})^M \leq (\frac{3}{4})^{\log_{4/3}(n)} = 1/n$.
   Then by the above formula the probability that quicksort takes at at most $\frac{8n}{3} \log_{4/3} n$ time is at at least $1 - 1/n$ (we can improve this some $c > 1$ and show $1 - 1/n^c$ probability by choosing a constant bigger than $8/3$)

3

1. In order to have $A\omega = \gamma$ mod 2, we need each have every element in the output vector of size m mod 2 to be the same as every corresponding element in γ:

   $P[A\omega[i] = \gamma[i] \text{ mod } 2]$ for all $0 \leq i \leq m-1$

   Since A and b is picked uniform at random, each element in A has probability exactly half of being 0 and exactly half of being 1. When you multiply matrix A with vector b, the index i of the result vector will be the vector multiplication of ith row in A and b. Since each element of the result vector has to mod 2, the index i of the result will based on the number of matching 1s in ith row in A and vector b(matching means in the same index of the vector b and the vector of ith row of A are both 1).
   Therefore, let k be the number of 1s in the vector b, $n-k$ will be the number of 0s in the vector. The probability of the result from vector b and the vector of ith row of A mod 2 being 0 (binomial theorem applies here):

   $$\frac{2^{(n-k)} \bullet \sum_{j=0}^{\lfloor (k/2) \rfloor} \binom{k}{2j}}{2^n} = \frac{\sum_{j=0}^{\lfloor (k/2) \rfloor} \binom{k}{2j}}{2^k} = \frac{2^{(k-1)}}{2^k} = \frac{1}{2}$$

   Then, the probability of the result from vector b and the vector of ith row of A mod 2 being 1:

   $$\frac{2^{(n-k)} \bullet \sum_{j=0}^{\lfloor (k-1/2) \rfloor} \binom{k}{2j+1}}{2^n} = \frac{\sum_{j=0}^{\lfloor (k-1/2) \rfloor} \binom{k}{2j+1}}{2^k} = \frac{1}{2}$$

   Therefore, if $\gamma[i] = 1$ for $0 \leq i \leq n-1$, $P[A\omega[i] \text{ mod } 2 = 1] = \frac{1}{2}$. If $\gamma[i] = 0$, $P[A\omega[i] \text{ mod } 2 = 0] = \frac{1}{2}$. Since $P[A\omega[i] = \gamma[i] \text{ mod } 2]$ is independent from $P[A\omega[j] = \gamma[j] \text{ mod } 2]$ if $i \neq j$, therefore, the $P[A\omega = \gamma \text{ mod } 2] = \prod_{i=1}^{m} P[A\omega[i] = \gamma[i]] = \frac{1}{2^m}$.

   In order to prove that for every X_u and X_v for $v \neq u$, they are independent. We need to show pairwise independent, which implies we have to show that $Pr[Au+b = \alpha \wedge Av+b = \beta] = Pr[Au+b = \alpha] \bullet Pr[Av+b = \beta]$ for any $v \neq u$.

   What we proved above, we can say that $Pr[Au+b = \alpha] = Pr[Av+b = \beta] = \frac{1}{2^m}$. Since

$Pr[Au + b = \alpha \wedge Av + b = \beta] = Pr[A(u - v) = (\alpha - \beta) \wedge b = \beta - Av]$, if we can show that $Pr[A(u-v) = (\alpha-\beta) \wedge b = \beta - Av] = Pr[A(u-v) = (\alpha-\beta)] \cdot Pr[b = \beta - Av]$, then we can calculate $Pr[Au + b = \alpha \wedge Av + b = \beta]$.

Because $Pr[Au = \alpha \wedge Av = \beta) = Pr[A(u - v) = (\alpha - \beta)] = \frac{1}{2^m}$(we can substitute the γ with $\alpha - \beta$ and $u - v$ with w), for any $b = \beta - Av$, the probability of $Pr[A(u - v) = (\alpha - \beta)] = \frac{1}{2^m}$. By the principle of conditional probability, if $P(A|B) = P(A)$ then A and B are independent. Therefore, $Pr[A(u-v) = (\alpha-\beta) \wedge b = \beta - Av] = Pr[A(u-v) = (\alpha-\beta)] \cdot Pr[b = \beta - Av]$ is true thus the probability of $Pr[A(u-v) = (\alpha-\beta) \wedge b = \beta - Av] = Pr[A(u-v) = (\alpha-\beta)] \cdot Pr[b = \beta - Av] = \frac{1}{2^m} \cdot Pr[b = \beta - Av]$. Once again, $Pr[b = \beta - Av] = Pr[Av = (\beta - b)] = \frac{1}{2^m}$.

Hence, $Pr[Au + b = \alpha \wedge Av + b = \beta] = Pr[A(u - v) = (\alpha - \beta)] \cdot Pr[b = \beta - Av] = Pr[A(u - v) = (\alpha - \beta)] \cdot Pr[b = \beta - Av] = \frac{1}{2^m} \cdot \frac{1}{2^m} = Pr[Au + b = \alpha] \bullet Pr[Av + b = \beta]$. We proved pairwise independent, then it is guaranteed that X_u and X_v are independent for $u \neq v$.

2. We proved that randomized vectors multiplication from size n to size n mod 2 has $\frac{1}{2}$ chance of being 1 and $\frac{1}{2}$ chance of being 0. Having 2 randomized vector of size n', where $n' \leq n$, it will still satisfies as long as vectors are random based on the calculation in part a (we never have specific restriction on the size).

   If a vector v of size n can divided into 2 parts and each part is a sub-vector of a randomized vector of size n, we can proved that $Pr[vb \bmod 2 = 1] = Pr[vb \bmod 2 = 0] = \frac{1}{2}$. Since we can divided b into same size of 2 sub-vector, the first sub-vector multiplies the first sub-vector of v and the second sub-vector multiplies the second sub-vector of v. The chance of first sub-vector multiplication has even 1s or odd 1s are both $\frac{1}{2}$, same for second sub-vector multiplication. Then, the probability of even number of 1s for vb is both sub-vector multiplication have odd 1s or both have even 1s, $\frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2}$. Therefore, the probability of having odd number of 1s is also $\frac{1}{2}$.
   We will divided the situation into 3 cases:

   (a) $m < n$: If we look at the last row of A, we can divided the row into two parts: first part of a randomized list of size s and one sub-vector of size $n - s$ come from row 1. Since the first row is selected at random, its sub-vector is also random. Then we know the last bit of Ab mod 2 will have $\frac{1}{2}$ be 1 and $\frac{1}{2}$ be 0. Similar for all the row between row 1 and the last row as they can be divided into one sub-vector of the first row and one sub-vector of the first part of the randomized list the last row, they are both random. Therefore, each index of $A\omega$ has probability of $\frac{1}{2}$ to be 1 after mod 2.

   (b) $m > n$: If m is slightly bigger than n, then the only change between $m < n$ is that the last row of A itself will be a random vector. Every row in between is the same, combination of one sub-vector from first row and one sub-vector from last row. Therefore, each index of $A\omega$ has probability of $\frac{1}{2}$ to be 1 after mod 2.

   (c) $m >> n$ Then the difference with m slightly larger than n is there will be multiple first rows and last rows. Every $n \times n$ size of matrix in A can be treat as $m > n$, therefore, every row is the combination of one sub-vector from its corresponding "first row" and one sub-vector from its corresponding "last row". Therefore, each index of $A\omega$ has probability of $\frac{1}{2}$ to be 1 after mod 2.

   Then we proved that every index of $A\omega$ has $\frac{1}{2}$ to be the same as γ.
   We also need to prove for every two row i and row j in A, where $i \neq j$, if we can prove that $Pr[A[i]b \bmod 2] = Pr[A[i]b \bmod 2 \mid A[j]b \bmod 2]$ for any $i \neq j$, then every two rows are independent. For simplicity, let's make row i in A R_i and row j in A R_j. As we are looking for matching 1s, we only pay attention to the index in b that is 1, these indexes will be same for R_i and R_j. Let k be the sum of indexes of 1 in b, we are choosing number 1s in these indexes in R_j and R_i: $Pr[R_jb$ mod

$2 == 0] = \frac{\sum_{j=0}^{\lfloor (k/2) \rfloor} \binom{k}{2j}}{2^k} = \frac{2^{(k-1)}}{2^k} = \frac{1}{2}$. Since we are looking over the same indexes for R_i and R_j, suppose for all x_j in these indexes in R_j has no intersection for all x_i in these indexes in R_i, then simply, the probability of $Pr[R_i b \bmod 2 == 0] = \frac{1}{2}$. If there are x_i that both in the indexes in R_i and R_j, seems we are shifting every row, x_i won't be in the same index for R_i and R_j, therefore, there will be some x_i are not in x_j. Suppose we have z number of duplicated in both indexes in R_i and R_j, we want to proved that what every number of 1s in these z duplicated x_i, the probability of $Pr[R_i b \bmod 2 == 0] = \frac{1}{2}$. If the duplicated number of z has odd 1s, $Pr[R_i b \bmod 2 == 0]$ under this condition, there should be odd 1s in the remaining of R_i to make the result mod 2 equal to 0. $Pr[R_i b \bmod 2 == 0] = \frac{\sum_{j=0}^{\lfloor (k-z)/2 \rfloor} \binom{(k-z)}{2j+1}}{2^{(k-z)}} = \frac{2^{(k-1-z)}}{2^{k-z}} = \frac{1}{2}$. When the duplicated number of z has even 1s, $Pr[R_i b \bmod 2 == 0]$ under this condition, there should be even 1s in the remaining of R_i to make the result mod 2 equal to 0. $Pr[R_i b \bmod 2 == 0] = \frac{\sum_{j=0}^{\lfloor (k-z)/2 \rfloor} \binom{(k-z)}{2j}}{2^{(k-z)}} = \frac{2^{(k-1-z)}}{2^{k-z}} = \frac{1}{2}$. Therefore, we proved that no matter how many the duplicated x_i are, the probability of R_i to be 0 and 1 are both 1/2 and it's not depended on whether R_j is 0 and 1. Therefore, by the principle of conditional probability, we can say that $Pr[A[i]b \bmod 2] = Pr[A[i]b \bmod 2 \mid A[j]b \bmod 2]$ for any $i \neq j$, thus every two rows are pairwise independent.

We will make an assumption for any $m \times n$ *Toeplitz* matrix A, the every outcome of $A\omega$ will be equally likely.
Base case: When $m = 1$, there is only one row, the outcome mod 2 is one single number. It has $\frac{1}{2}$ chance of being 1 and $\frac{1}{2}$ chance of being 0, therefore, among all outcome, each outcome has same probability.
Induction step: Suppose it's true for $k \times n$, we want to prove that it holds for $(k+1) \times n$ still holds. Since we proved before $k \times n$, each outcome will have single probability in $k \times n$, when adding one row, the row itself will have $\frac{1}{2}$ probability of being 1 and $\frac{1}{2}$ of being 0. Since the row is pairwise independent to every row above, any outcome's of any row above will not be affect, and the outcome of $k+1$ row will still be uniformly distributed($\frac{1}{2}$ being 0 and $\frac{1}{2}$ being 1).
Therefore, the $(k+1) \times n$ *toeplitz* will have every outcome uniformly distributed. Hence, we proved that for any *toeplitz* A matrix of $m \times n$, each outcome is uniformly distributed. Therefore, with total number of outcomes are 2^m, the probability of $P[A\omega = \gamma \bmod 2] = \frac{1}{2^m}$.

3. The number of bits we need to generated the *Toeplitz* matrix $A \in \{0,1\}^{m \times n}$ is $logM + logN - 1$. The storage is also $logM + logN - 1$.

4 X_i be the value of the counter after i events, and $Y_i = (1+a)^{X_i}$. For $n = 0, 1$ $Y_i = 1, 1+a$ deterministically. $\mathbb{E}(Y_n) = an + 1$ Proof by induction on n

Proof:

$$\begin{aligned}
\mathbb{E}(Y_n) &= \mathbb{E}((1+a)^{X_n}) \\
&= \sum_{j=0}^{\infty} (1+a)^j \Pr(X_n = j) \\
&= \sum_{j=0}^{\infty} (1+a)^j (\Pr(X_{n-1} = j) \cdot (1 - \frac{1}{(1+a)^j}) + \Pr(X_{n-1} = j-1) \cdot \frac{1}{(1+a)^{j-1}}) \\
&= \mathbb{E}(Y_{n-1}) + \sum_{j=0}^{\infty} (1+a) \Pr(X_{n-1} = j-1) - \Pr(X_{n-1} = j) \\
&= \mathbb{E}(Y_{n-1}) + a \\
&= a(n-1) + 1 + a && \text{(By induction)} \\
&= an + 1
\end{aligned}$$

■

So the estimate for n the algorithm outputs is $\frac{(1+a)^X - 1}{a}$
$\mathbb{E}(Y_n^2) = an(a+2)(a(n-1)+2)/2 + 1$. $Y_n^2 = 1, (1+a)^2$ determinstically for $n = 0, 1$.

Proof:

$$\begin{aligned}
\mathbb{E}(Y_n^2) &= \mathbb{E}((1+a)^{2X_n}) \\
&= \sum_{j \geq 0} (1+a)^{2j} \Pr(X_n = j) \\
&= \sum_{j \geq 0} (1+a)^{2j} (\Pr(X_{n-1} = j) \cdot (1 - \frac{1}{(1+a)^j}) + \Pr(X_{n-1} = j-1) \cdot \frac{1}{(1+a)^{j-1}}) \\
&= \mathbb{E}(Y_{n-1}^2) + \sum_{j \geq 0} (1+a)^{j+1} \Pr(X_{n-1} = j-1) - (1+a)^j \Pr(X_{n-1} = j) \\
&= \mathbb{E}(Y_{n-1}^2) + (a^2 + 2a)\mathbb{E}(Y_{n-1}) \\
&= \mathbb{E}(Y_{n-1}^2) + (a^2 + 2a)(an - a + 1) \\
&= an(a+2)(a(n-1)+2)/2 + 1 && \text{(By induction)}
\end{aligned}$$

■

$\mathrm{Var}(Y_n) = \frac{a^3 n}{2}(n-1)$ and $\mathrm{Var}(\tilde{n}) = \frac{an}{2}(n-1)$ By applying Chebyshev we get

$$\begin{aligned}
\Pr(|\tilde{n} - n| \geq \epsilon n) &\leq \frac{an}{2n^2\epsilon^2}(n-1) \\
&\leq \frac{a}{2\epsilon^2}(1 - 1/n) \\
&\leq \frac{a}{2\epsilon^2} \\
&\leq 1/10 && \text{(this is true when } a \leq \epsilon^2/5)
\end{aligned}$$

This implies that for $0 < a \leq \epsilon^2/5$, $\Pr(|\tilde{n} - n| \leq \epsilon n|) \geq 9/10$
The number of bits the algorithm uses is $O(\log X)$, where X is the value of the counter after n increments.
The previous part shows that $\tilde{n} \leq n(1+\epsilon)$ with probability at least 9/10.

$$\begin{aligned}
\frac{(1+a)^X - 1}{a} &\leq (1+\epsilon)n \\
X \log(a+1) &\leq \log(an(1+\epsilon)+1) \\
X &\leq \frac{\log(an(1+\epsilon)+1)}{\log(a+1)} \\
&= \log_{a+1}(an(1+\epsilon)+1)
\end{aligned}$$

Therefore, $S(n) = \log(\log_{a+1}(an(1+\epsilon)+1))$ with probability at least 9/10, if $\epsilon \leq 1$ then $O(\log(\log n))$ bits are used.

Algorithm 1 Randomized Quicksort

Require: A is the array of size n,lo is the lowest index and hi is the highest index.

RandomizedQuicksort(A, lo,hi):
$n \leftarrow hi - lo + 1$
$i \leftarrow random(lo...hi)$
Count $\leftarrow 0$
for $j \leftarrow lo$ to hi **do**
  if $A[j] < A[i]$ **then**
    Count = Count +1
  end if
end for
while $Count < n/4 \parallel Count > 3n/4$ **do**
  $i \leftarrow random(lo..hi)$
  for $j \leftarrow lo$ to hi **do**
    if $A[j] < A[i]$ **then**
      Count = Count +1
    end if
  end for
end while
$temp \leftarrow A[Count]$
$A[Count] \leftarrow A[i]$
$A[i] \leftarrow temp$
$leftstart \leftarrow lo$
$rightstart \leftarrow Count + 1$
while $start < Count$ **do**
  if $A[leftstart] > A[Count]$ **then**
    $temp \leftarrow A[leftstart]$
    $A[leftstart] \leftarrow A[rightstart]$
    $A[rightstart] \leftarrow temp$
    $rightstart = rightstart + 1$
  else
    $leftstart = leftstart + 1$
  end if
end while
$RandomizedQuicksort(A, lo, Count - 1)$
$RandomizeQuicksort(A, Count + 1, hi)$