NAME: VIJETA JOHRI

ANS 1 The recursion tree for quicksort is given as follows. Note in the tree $\alpha$ is represented as $a$.

Figure 1: Recursion tree for Quicksort
Let minimum depth be given by i

\[ n\alpha^{i} = 1 \]
\[ i = -\log_{\alpha} n \]
\[ = -\frac{\ln n}{\ln \alpha} \]

Let maximum depth be given by k

\[ n(1 - \alpha)^{k} = 1 \]
\[ k = -\log_{1-\alpha} n \]
\[ k = -\frac{\ln n}{\ln(1-\alpha)} \]

**ANS 2** Insertion sort and merge sort are stable while quicksort is unstable. Insertion sort is stable as in the lines when we try to insert \( a[j] \) into sorted sequence \( a[1..j-1] \):

key \( = \) \( a[j] \)

\( i = j - 1 \)

while \( i > 0 \) and \( A[i] > \text{key} \)

\{

\( a[i+1] = a[i] \)

\( i = i - 1 \)

\}

\( a[i+1] = \text{key} \)

In these lines \( a[i] \) is not shifted to \( a[i+1] \) if \( a[j] \) is equal to \( a[i] \). So the relative ordering is maintained.

In mergesort in order to merge two lists we pick the smaller of the two numbers in the front of the list and store it into a new list. If the two numbers are equal we pick the number from the first list. So relative ordering is maintained. Quicksort is unstable as due to the exchange process in the partition function it is possible that a number that used to come after another equal number will be placed before that number. So the relative ordering is disturbed in the new list and we have no way to tell what was the relative ordering of the two numbers in the original sequence. In order to fix the problem we can include a sequence number along with the number denoting the relative ordering among the various equal numbers. If initially we were using \( A[1..n] \) to store the original sequence now we can use a double dimensional array \( B[1..n][1..2] \). \( B[i][1] \) will store the value \( A[i] \) and \( B[i][2] \) will store the sequence number of \( A[i] \). So the revised quicksort algorithm is

\[ \text{Quicksort}(B,p,r) \]
\[ \{ \]
\[ \text{if}(p < r) \]
\[ \{ \]
\[ q = \text{partition}(B,p,r) \]
\[ \text{Quicksort}(B,p,q) \]
\[ \text{Quicksort}(B,q+1,r) \]
\[
E[X] = \sum_{i=0}^{\infty} i \Pr \{ X = i \} \\
= \sum_{i=1}^{\infty} i (\Pr \{ X \geq i \} - \Pr \{ X \geq i + 1 \}) \\
= \Pr \{ X \geq 1 \} - \Pr \{ X \geq 2 \} + 2 \Pr \{ X \geq 2 \} - 2 \Pr \{ X \geq 3 \} + 3 \Pr \{ X \geq 3 \} - 3 \Pr \{ X \geq 4 \} + \ldots \\
= \sum_{i=1}^{\infty} \Pr \{ X \geq i \} \\
= \sum_{i=1}^{t-1} \Pr \{ X \geq i \} + \Pr \{ X \geq t \} + \sum_{i=t+1}^{\infty} \Pr \{ X \geq i \} \\
\geq \sum_{i=1}^{t-1} \Pr \{ X \geq i \} + \Pr \{ X \geq t \} \\
= \Pr \{ X \geq 1 \} + \Pr \{ X \geq 2 \} + \ldots \Pr \{ X \geq t \} \\
\geq \Pr \{ X \geq t \} + \Pr \{ X \geq t \} + \ldots \Pr \{ X \geq t \} \text{ for } t > 0 \\
\text{(since } \Pr \{ X \geq i \} \geq \Pr \{ X \geq t \} \text{ for all } i \text{ where } 1 \leq i \leq (t-1)) \\
= \Pr \{ X \geq t \} \\
E[X]/t \geq \Pr \{ X \geq t \}
\]

ANS 4 Let T(n) be the best case running time for quicksort on an input of size n.
\[
T(n) = \min_{1 \leq q \leq n-1} \left( T(q) + T(n - q) \right) + \Theta(n)
\]
Let us guess that \( T(k) \geq c k \lg k \) for some constant \( c \).

Basis:
\( T(2) \geq 2 \lg 2 \) if we choose some particular \( c \) this inequality will hold

So let us assume \( T(k) \geq c k \lg k \)

On substituting this value of \( T(k) \) we get
\[
T(n) \geq c \min_{1 \leq q \leq n-1} \left( q \lg q + (n - q) \lg (n - q) \right) + \Theta(n)
\]
Let us find the minima of expression \( q \lg q + (n - q) \lg (n - q) \) for \( 1 \leq q \leq (n-1) \)
\[
f(q) = q \lg q + (n - q) \lg (n - q)
\]
3
for minima \( \frac{df}{dq} = 0 \)
\[ \frac{df}{dq} = 1 + lgq - lg(n - q) - \frac{(n - q)}{(n - q)} = 0 \]
\[ lgq = lg(n - q) \]
\[ lg\left(\frac{n - q}{n - q}\right) = 0 \]
\[ q = n - q \]
\[ q = n/2 \]
Putting this value of \( q \) we get
\[ T(n) \geq c[(n/2)lg(n/2) + (n/2)lg(n/2)] + \Theta(n) \]
\[ = cnlg(n/2) + \Theta(n) \]
\[ = cn(lgn - 1) + \Theta(n) \]
\[ = cnlgn - cn + \Theta(n) \]
\[ \geq cnlgn \]
Since we can pick \( c \) such that \( cn \) is less than \( \Theta(n) \)
So \( T(n) = \Omega(nlgn) \)
Hence proved.

**ANS 5** Let us define a random variable \( X \) to be the number of empty bins after \( n \) tosses.
Let \( X_i \) be the random variable whose value is 1 if the bin \( i \) is empty after \( n \) tosses and 0 if the bin is not empty.
So \( X = X_1 + X_2 + ..... X_n \)
So the expected number of empty bins will be
\[ E[X] = E[X_1 + X_2 + .. + X_n] \]
\[ = E[X_1] + E[X_2] + ...E[X_n] \ (1) \]
So \( E[X_i] = \sum x Pr\{X = x\} = 1.Pr\{X_i = 1\} + 0.Pr\{X_i = 0\} \]
\[ = Pr\{X_i = 1\} \]
\( Pr\{X_i = 1\} \) is the probability that a particular bin \( i \) is empty after \( n \) tosses.
The probability that a ball will land in any given bin \( i \) is \( \frac{1}{n} \)
The probability that a ball will not land in the given bin \( i \) or that the bin \( i \) will remain empty after one toss is \( 1 - \frac{1}{n} \)
The probability the bin \( i \) will remain empty after \( n \) tosses is \( = Pr\{X_i = 1\} = \left(1 - \frac{1}{n}\right)^n \)
So from \( (1) \) \[ E[X] = E[X_1] + E[X_2] + ...E[X_n] \]
\[ = \left(1 - \frac{1}{n}\right)^n + \left(1 - \frac{1}{n}\right)^n + .......(1 - \frac{1}{n})^n \ (n \ times) \]
\[ = n\left(1 - \frac{1}{n}\right)^n \]
So expected number of empty bins after \( n \) tosses is \( = n\left(1 - \frac{1}{n}\right)^n \)
Let us now define random variable \( X \) to be the number of bins with 1 ball after \( n \) tosses.
Let \( X_i \) be the random variable whose value is 1 if the bin \( i \) has 1 ball after \( n \) tosses and 0 otherwise.
So \( X = X_1 + X_2 + ..... X_n \)
So the expected number of bins with exactly one ball will be
\[ E[X] = E[X_1 + X_2 + \ldots + X_n] \]
\[ = E[X_1] + E[X_2] + \ldots E[X_n] \]
So \[ E[X_i] = \sum x Pr\{X = x\} = 1Pr\{X_i = 1\} + 0Pr\{X_i = 0\} \]
\[ = Pr\{X_i = 1\} \]
\(Pr\{X_i = 1\}\) is the probability that a particular bin i has exactly one ball after n tosses.
The probability that a ball will land in any given bin i is \(\frac{1}{n}\)
The probability that a ball will not land in the given bin i or that the bin i will remain empty after one toss is \(1 - \frac{1}{n}\)
The probability the bin i will have one ball after n tosses is \(n \left(1 - \frac{1}{n}\right)^{(n-1)}\) (as we can choose one bin out of the n bins in n ways)
\[ Pr\{X_i = 1\} = \left(1 - \frac{1}{n}\right)^{(n-1)} \]
So from (1) \[ E[X] = E[X_1 + X_2 + \ldots E[X_n] \]
\[ = \left(1 - \frac{1}{n}\right)^{(n-1)} + \left(1 - \frac{1}{n}\right)^{(n-1)} + \ldots + \left(1 - \frac{1}{n}\right)^{(n-1)} \] (n times)
\[ = n\left(1 - \frac{1}{n}\right)^{(n-1)} \]
**ANS 6** Let \(E\) be the event that we succeed in hiring the best qualified applicant.
Let \(E_i\) be the event that we hire the best qualified applicant on hiring the candidate in position i.
\[ Pr\{E\} = \sum_{i=1}^{n} Pr\{E_i\} \]
As we will never hire any applicant in positions from 1 to k \(Pr\{E_i\}=0\) for \(1 \leq i \leq k\)
So we get \[ Pr\{E\} = \sum_{i=(k+1)}^{n} Pr\{E_i\} \]
For event \(E_i\) to occur we need two events to occur:
The best applicant be in position i \(- B_i \)
All the applicants from position \(k+1\) to \(i-1\) should be lesser qualified than the most qualified candidate within the first k persons interviewed \(- L_i \)
\[ Pr\{B_i\} = \frac{1}{n} \]
\[ Pr\{L_i\} = \frac{k}{i-1} \]
assuming that qualification is a numerical quantity and all qualifications are distinct
\[ Pr\{E_i\} = Pr\{L_i \cap B_i\} \]
\[ = Pr\{B_i\} Pr\{L_i\} = \frac{k}{n(i-1)} \]
( since events \(B_i\) and \(E_i\) are independent)
\[ Pr\{E\} = \sum_{i=(k+1)}^{n} Pr\{E_i\} = \sum_{i=(k+1)}^{n} \frac{k}{n(i-1)} \]
\[ = \frac{k}{n} \sum_{i=(k+1)}^{n} \frac{1}{i-1} \]
\[ = \frac{k}{n} \sum_{i=k}^{n-1} \frac{1}{i} \]
We approximate by integrals to bound this sum from above and below.
\[ \int_{k}^{n} \frac{1}{x} dx \leq \sum_{i=k}^{n-1} \frac{1}{i} \leq \int_{k-1}^{n-1} \frac{1}{x} dx \]
\[ \frac{k}{n}(lnn - lnk) \leq Pr\{S\} \leq \frac{k}{n}(ln(n-1) - ln(k-1)) \]
As we wish to maximize the probability let us maximize the lower bound.
\[ f(k) = \frac{k}{n}(lnn - lnk) \]
for maxima $\frac{df}{dk} = 0$
On differentiating we get
\[ \frac{1}{n}(lnn - lnk - 1) = 0 \]
\[ lnk = lnn - 1 \]
\[ lnk = ln(n/e) \]
\[ k = n/e \]
Putting this value of $k$ in lower bound we get
\[ f(k) = \frac{k}{n}(lnn - lnk) \]
\[ f(n/e) = \frac{1}{e} \]
So professor dixon’s chances of hiring the best qualified applicant is approximately $1/e$