Homework 1
CS 7530 Solutions
Randomized Algorithms

Problem 1

Procedure `Permute`(*A[1..n]*)

> **for** $i = 1$ *to* n **do**
> > $b_i \in_R \{0, 1\}$;
> > **if** $b_i = 0$ **then**
> > > $B[j++] = A[i]$;
> >
> > **if** $b_i = 1$ **then**
> > > $C[k++] = A[i]$;
>
> Return Permute(B) $\cdot$ Permute(C);

Let π be the premutation generated by the algorithm. We want to show that π is a random permutation. Clearly each permutation is generated with a positive probability. It is enough to show that for all permutations σ and for all $i, j \in [n], i \neq j$, the probability that $\pi(i) = \sigma(i)$ and $\pi(j) = \sigma(j)$, conditioned on the event that $\pi(k) = \sigma(k)$ for all $k \neq i, j$, is 0.5. (**Exercise:** Prove this.) [1]

W.l.o.g let $\sigma(i) < \sigma(j)$. Consider the first iteration in which $b_i \neq b_j$. Conditioned on the event that $\pi(k) = \sigma(k)$ for all $k \neq i, j$, $\pi(i) = \sigma(i)$ if and only if $b_i = 0$, which happens with probability 0.5.

The analysis of the running time is similar to that of the quicksort. The number of random bits used is bound by the running time.

Problem 2

Consider the graph on vertices $\{s, t, v_1, v_2, \ldots, v_n\}$ with the edges $\cup_{i=1}^{n} \{(s, v_i), (v_i, t)\}$. Then the number of min-cuts separating s and t are 2^n where the graph has $n + 2$ edges. In fact, all cuts separating s and t are min-cuts.

Consider the graph as above and "double" every edge on the s-side, by adding a parallel edge (s, v_i) for all i. Now the only min-cut is $(\{s, v_1, v_2, \ldots, v_n\}, \{t\})$, which the algorithm finds if at every step it chooses an (s, v_i) edge. This happens with probability $2/3$, and hence the probability of finding a min-cut is $\left(\frac{2}{3}\right)^n$.

[1] A lot of you have not proved that the permutation generated is uniformly at random; instead, a weaker statement is proved. Consider, for instance, the permutation of numbers modulo n, given by picking a number a randomly, and sending x to $x + a \bmod n$ and see if this algorithm violates your statement.

Problem 3

We will prove all 3 parts by induction on the height h of the tree. Let 1,2 and 3 be the children of the root.

Part a

Suppose $h = 1$. Suppose that a deterministic algorithm D does not read leaf 3, w.l.o.g. Then it cannot distinguish between (0,1,0) and (0,1,1). Hence D reads all 3 leaves. Now let $h > 1$. Suppose that D cannot determine the value of 3, w.l.o.g. Then as before it fails on some input. Hence D has to find the value of 1,2 and 3. By induction, it reads 3^h leaves.

Part b

We will prove the inductive case only, the base case is similar. W.l.o.g suppose that 1 and 2 have the same value. Consider the non-deterministic algorithm N that recursively determines the value of 1 and 2 and outputs that value. This reads only 2^n leaves.

Part c

The inductive hypothesis here is that the expected number of leaves read on a tree of height h is $\left(\frac{8}{3}\right)^h \leq 3^{0.9n}$. Again, inductive case only. Again w.l.o.g suppose that 1 and 2 have the same value. Then w.p at least 1/3, R picks 1 and 2. $E[\text{number of leaves read}] \leq (2 \cdot 1/3 + 3 \cdot 2/3)\frac{8}{3}^{h-1} = \left(\frac{8}{3}\right)^h$.

Problem 4

Suppose you query N residents uniformly and independently at random. Let X be the number of republicans among these. The estimate $\hat{f} := X/N$. By linearity of expectations, $\mu := E[X] = fN \geq aN$.

$$\begin{aligned}
\mathbf{Pr}[|f - \hat{f}| \geq \epsilon f] &= \mathbf{Pr}[|X - \mu| \geq \epsilon\mu] \\
&\leq 2\exp\left(\frac{-\mu\epsilon^2}{3}\right) \text{ by chernoff bounds,} \\
&\leq 2\exp\left(\frac{-aN\epsilon^2}{3}\right), \\
\text{which is at most } \delta \text{ if we choose } N &\geq \frac{3\log(2/\delta)}{\epsilon^2 a}.
\end{aligned}$$

1 Problem 5

Define the random variables

$$X_i = \begin{cases} 1 & \text{if a 6 comes up in the } i\text{th throw of the die,} \\ 0 & \text{otherwise.} \end{cases}$$

$X = \sum_i X_i$. $\mu := E[X] = n/6$. $p = \mathbf{Pr}[X > n/4]$. Using Markov's Inequality, we get

$$p < \frac{\mu}{n/4} = 2/3.$$

$\mathbf{Var}(X_i) = E[X_i^2] - E^2[X_i] = 1/6 - 1/36 = 5/36$. $\mathbf{Var}(X) = \sum_i \mathbf{Var}(X_i) = 5n/36$. By Chebyshev's Inequality, we get

$$p \leq \mathbf{Pr}[|X - \frac{n}{6}| \geq \frac{n}{12}] \leq \frac{5n/36}{(n/12)^2} = \frac{20}{n}.$$

By Chernoff bounds,

$$p = \mathbf{Pr}[X - \frac{n}{6} \geq \frac{n}{12}] = \mathbf{Pr}[X - \mu \geq \frac{1}{2}\mu] \leq \exp\left(\frac{-\mu(1/2)^2}{3}\right) = \exp\left(\frac{-n}{72}\right).$$

Problem 6

Let $\delta(t) = \mathbf{Pr}[\text{at most } k \text{ copies of coupon 1 are collected at time } t]$. [2]

Lemma 1. *If $\delta \leq 1/2n$ then the expected time to get $k+1$ copies of all n coupons $\leq 2t$.*

Proof. Define the random variables

$$X_i = \begin{cases} 1 & \text{if at most } k \text{ copies of coupon } i \text{ are collected at time } t, \\ 0 & \text{otherwise.} \end{cases}$$

Let $X = \sum_i X_i$. Then $E[X] = \delta n \leq 1/2$. Therefore by Markov's inequality, $\mathbf{Pr}[X < 1] \geq 1/2$. $X < 1$ implies we have at least $k+1$ copies of all coupons at time t. The lemma follows. □

Let

$$\begin{aligned} P(r,n,t) &:= \mathbf{Pr}[\text{exactly } r \text{ copies of coupon 1 are collected at time } t]. \\ &= \binom{t}{r}(1/n)^r p^{t-r} \text{ (where } p = 1 - 1/n) \\ &\leq p^t \left[\frac{te}{nrp}\right]^r \\ &\sim e^{-\alpha}\left[\frac{e\alpha}{rp}\right]^r \end{aligned}$$

Where $\alpha := t/n$ and we have used the approximation that $(1 - 1/n)^n \sim e^{-1}$. Now

$$\delta = \sum_{r=0}^{k} P(r,n,t) = e^{-\alpha}\sum_{r=0}^{k}\left[\frac{e\alpha}{rp}\right]^r.$$

Verify that there is a constant c such that if $\alpha = \log n + k \log\log n + c$ then $\delta \leq 1/2n$.

[2]We will just write δ from here on, and it is to be understood that it is a function of t.

Problem 7

Whenever there is a good node in a path from the tree to a leaf, the size of the list decreases by a factor of 2/3 or less. The size of the list at the root is n and if there are t good nodes, then the size of the list at the leaf is at most $n\left(\frac{2}{3}\right)^t$. But the size of the list at the leaf, by definition, is 1. Hence $n\left(\frac{2}{3}\right)^t \geq 1 \Rightarrow t \leq c \log n$ where $c = \frac{1}{\log(3/2)}$.

A good node in a list of size n is one whose rank is between $n/3$ and $2n/3$. Hence the probability of choosing a good node as a pivot is $1/3$. Let

$$X_i = \begin{cases} 1 & \text{w.p } 1/3, \\ 0 & \text{w.p } 2/3, \end{cases}$$

for $i = 1, \ldots, t$. Let $X = \sum_i X_i$. Then $E[X] = t/3$. Consider any path of the tree from the root to the leaf. We know from the previous part that the number of good nodes on it is at most $c \log n$. Hence $\mathbf{Pr}$[the path has length at least t] $\leq \mathbf{Pr}[X < c \log n]$. By Chernoff bounds, $\mathbf{Pr}[X < c \log n] \leq \exp(-(E[X] - c \log n)^2 / 2E[X]) = \exp(-3(t/3 - c \log n)^2 / 2t)$. Verify that by choosing $t = c' \log n$ for some constant c', one can ensure that $\mathbf{Pr}[X < c \log n] \leq n^{-2}$.

We showed that the probability that any particular path from the root to a leaf is longer than $c' \log n$ is at most n^{-2}. Since there are n leaves, there are n such paths. By union bound the probability that some such path is longer than $c' \log n$ is at most $1/n$. In other words, the probability that all paths are shorter than $c' \log n$ is at least $1 - 1/n$.

The running time of the quicksort algorithm is bounded by the length of the longest path times n. Hence with probability $1 - 1/n$ it runs in time $\leq c' \log n$.