CS648: Randomized Algorithm
Assignment – I
Ashwani Kumar Gautam(13165)

Problem 2.

Strategy:

1. Pick at random, a number from 0 to n-1, lets call it a .
2. Pick $(z-a) \bmod n$.
3. Lookup value of a and $(z-a) \bmod n$ from the table given and evaluate
4. Repeat steps 1-3, *k* times for lower probability of failure.
5. Record all the *k* output obtained after step 4.
6. Output the value which has maximum frequency of occurrence.

Analysis:

We try to analyze the error probability, that is, possibility of not getting true $F(z)$.

Its obvious that if F(a) and F(z-a) were not changed previously, then we would get the correct output.

So we could get wrong output only for the below mentioned conditions:

1. Value of $F(a)$ or $F((z-a) \bmod n)$ was changed.
2. Values of both $F(a)$ or $F((z-a) \bmod n)$ were changed.

Note that, the output of function goes through a modulus of m, we still have the chances of colliding with the correct answer even if $F(a)$ and/or $F((z-a) \bmod n)$ is/are changed.

Thus,

$$Pr\{Wrong\,Ouput\} \le Pr\{F(a)changed\} \cup Pr\{F(z-a)changed\} \le Pr\{F(a)changed\} + Pr\{F(z-a)changed\}$$

$$Pr\{F(a)changed\} = \frac{1}{5} = Pr\{F(z-a) \bmod n\, changed\}$$

$$Pr\{Wrong\,Output\} \le \frac{1}{5} + \frac{1}{5} = \frac{2}{5}$$

Thus,

$$Pr\{Correct\,output\} \ge \frac{3}{5} > \frac{1}{2}$$

Repeat the above algorithm *k* times and record all the values of F(z) obtained.

Since each run of the algorithm is independent of the previous run and the future run, all the iteration are mutually independent.

Three cases can occur now

Case 1 : All k values of F(z) are identical.

If this is the case, then probability of error arises from the fact that we were unlucky all k times. Thus,

$$Pr\{Wrong\,Output\} \leq \left(\frac{2}{5}\right)^{k}$$

which is hopelessly low as k increases

Case 2: All values of F(z) are different.

If this is the case then, we are no better than our first iteration and we can simply output any one of the k value we have for F(z).

with

$$Pr\{Wrong\,Output\} \leq \frac{2}{5}$$

Case 3: Neither all are different nor all are identical.

If this is the case then we can simply output the value F(z) which occurs most number of times(not equal to k, obviously). Suppose the most frequent value occurs m times, then

$$Pr\{Wrong\,Output\} \leq \left(\frac{2}{5}\right)^{m} < \frac{2}{5}$$

Thus, for any case we have,

$$\left(\frac{2}{5}\right)^{k} \leq Pr\{Wrong\,Output\} \leq \frac{2}{5}$$

Problem 3.

N $\rightarrow$ Total number of applicants.
W $\rightarrow$ Event that best candidate is selected.

Then,

$$Pr\{W\}=\sum_{i=1}^{N} Pr\{W_i\}$$

Here, W_i is the event that i^{th} was the best candidate and was selected.
Clearly all W_i are disjoint and hence the equality in the above equation.

Our sample space consists of all the N! orderings of candidates, we will **partition** the sample space based on where the overall best candidate appears. Clearly this scheme divides the sample space into N **mutually exclusive and exhaustive** events, because the candidate cannot simultaneously be at two different position.

Since, Google rejected the first *k* applicants after interviewing them, we are only interested in

$$Pr\{W\}=\sum_{i=k+1}^{N} Pr\{W_i\} \quad \text{. . . .(a)}$$

Event W_i can happen only if both the following conditions hold:

1. Best candidate is interviewed at i^{th} position.
2. No candidate between *k+1* to *i-1* is hired.

Lets call condition 1 and 2 to be event P_i and Q_i respectively.

Clearly Event P_i and Q_i are independent. Event Q_i is depends only on the order of candidate from 1 to *i*-1, whereas even P_i depends only if the i^{th} candidate is the best among all, and have no influence on the ordering of candidate from 1 to *i-1*.

Thus following independence of two event, we notice

$$Pr\{W_i\}=Pr\{P_i\}*Pr\{Q_i\}$$

Thus (a) becomes,

$$Pr\{W\}=\sum_{i=k+1}^{N} Pr\{P_i\}*Pr\{Q_i\} \quad \text{…..(b)}$$

Since the best candidate can be placed anywhere in random permutation,

$$Pr\{P_i\}=\frac{1}{N}$$

For Q_i to occur, the best candidate out of the first i-1 candidates has to be one of k candidate which were interviewed first. Thus we need to determine the probability of occurrence of best candidate till i -1 in the first k positions, which is simply

$$Pr\{Q_i\}=\frac{k}{i-1}$$

Thus, from (b)

$$Pr\{W\}=\sum_{i=k+1}^{N}\frac{k}{N(i+1)}=\frac{k}{N}\sum_{i=k+1}^{N}\frac{1}{i-1}$$

$$=\frac{k}{N}\sum_{i=k}^{N-1}\frac{1}{i}$$

Approximating the last integral, we have

$$\int_k^N\frac{1}{x}dx\leq\frac{k}{N}\sum_{i=k}^{N-1}\frac{1}{i}$$

which on integration gives,

$$\frac{k}{N}(\ln(N)-\ln(k))\leq Pr\{W\}$$

Differentiating the lower bound with respect to k,

$$\frac{1}{N}(\ln(N)-\ln(k)-1)$$

for,

$$\ln(N)=1-\ln(k) \quad => \quad k=\frac{N}{e}$$

the derivative reaches zero and thus lower bound reaches its maximum.

Thus for $k=\frac{N}{e}$ Google have the maximum probability of hiring, with

$$Pr\{W\}\geq\frac{N}{e}(\ln(N)-\ln(\frac{N}{e}))=\frac{1}{e}$$