CS 344 Assignment 4

November 13, 2018

Name : Yang Bao
NetID : yb184
Section : 02
The classmate I discussed with : Zitian Qin, Hao Jiang

Problem 1 :

1. Part 1 :
   Since we assume that $A[j] = k$ we can have that $E[X_1] = \frac{1}{j}$

2. Part 2 :
   $E[X_i] = \frac{1}{j-i+1}$

3. Part 3 :
   Expected running time is $\sum_{i=1}^{n} \frac{1}{j-i+1}$.
   Since j is a constant although we don't know what exactly it is, we can rewrite the formula into :
   $\sum_{i=1}^{n} \frac{1}{j-i+1} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} = O(logn)$

4. Part 4 :
   Since the constant j as $A[j] = k$ doesn't actually influence the expected running time in the previous 3 question but only plays a role as an index to check if our pivot is larger or smaller than the target, we can define j in a new way. As k is no longer in the array, we can define j as $A[j]$ is the element which is most close to k in the array. No matter $A[j]$ is larger or smaller than k, its role as the index will not change and doesn't influence the expected running time.

Problem 2 :

1. Part 1 :
   Let $k = 1$ and $k' = 2$, for any $b \geq 2$ we have $h(k) = 1$ and $h(k') = 1$ so $h(k) = h(k')$ and so k and k' collide. Then the probability of collision is $\frac{n-1}{n}$ which is much bigger than the expected $\frac{1}{n}$. So that is not a universal hash function.

2. Part 2 :
   The Set S I constructed is:
   $S = \{1, 1 + n \cdot (\frac{n}{3})!, 1 + 2n \cdot (\frac{n}{3})!, 1 + 3n \cdot (\frac{n}{3})!, \ldots, 1 + \frac{n}{2} \cdot n \cdot (\frac{n}{3})!, 2, 3, \ldots, \frac{n}{2}\}$
   Let me explain the set S, the first $\frac{n}{2}$ of S can be concluded into a formula that $1+t\cdot n\cdot(\frac{n}{3})!$ where $0 \leq t \leq \frac{n}{2}$ and the rest elements are just $\frac{n}{2}$ integers from 2 to $\frac{n}{2}$
   Now consider the first part of S, we can see that :
   $h(1 + t \cdot n \cdot (\frac{n}{3})!) = \lceil \frac{1+t\cdot n\cdot(\frac{n}{3})!}{b} \rceil \mod n$
   $= \lceil \frac{1}{b} \rceil \mod n + \lceil \frac{t\cdot n\cdot(\frac{n}{3})!}{b} \rceil \mod n$
   $= 1 + \lceil \frac{t\cdot n\cdot(\frac{n}{3})!}{b} \rceil \mod n$
   When $1 \leq b \leq \frac{n}{2}$, $\lceil \frac{t\cdot n\cdot(\frac{n}{3})!}{b} \rceil \mod n = 0$, so $h(1 + t \cdot n \cdot (\frac{n}{3})!) = 1$, which means that all those $\frac{n}{2}$ elements collide.
   We don't know for the rest $\frac{2}{3}$ b if those elements collide but it's safe to say that for the S I constructed, there is at least $\frac{1}{3}$ probability that $\frac{n}{2}$ keys collide in the same spot 1.

3. Part 3 :
   According to part 2, we know that there will be $\frac{1}{3}$ probability that $\frac{n}{2}$ keys collide in the same spot, in the worst case, we search for those element which are collide, we will need $\frac{n}{2}$ time. So the big-O is $O(n)$.

Problem 3 :

1. Part 1 :
   Since the images are made up with $30 \times 30 = 900$ pixels and every pixel can either black or white, which is 2 choices, so the amount of all possible images is 2^{900} which is also the size of universe U.

2. part 2 :
   Let I_1 be an image that all the pixels are black and I_2 be an image that only 1 pixel is white and all other 899 pixels are black. Then unless the 20 pixels chose for I_2 contain the white pixel, $h(I_1) = h(I_2)$
   The probability that the white pixel being chose is $\frac{1 \cdot \binom{899}{19}}{\binom{900}{20}} = \frac{\frac{899!}{19!880!}}{\frac{900!}{20!880!}} = \frac{1}{45}$.
   So $Pr[h(I_1) = h(I_2)] = 1 - \frac{1}{45} = \frac{44}{45}$, which is significantly larger than $\frac{1}{2^{20}}$. So h is not universal.

Problem 4 :
Before writing my pseudocode I assume that the Universal Hash Function you give us on page 6 from the lecture note for lecture 6 is an universal hash function and I can use that and I will refer that function as $h()$.

```
Declare an arraylist S of type index of size n          O(1)
for i in (0,n-1){                                        O(n)
    S[ h( A[i] ) ] = i
}
int min = inf, a = 0, b = 0, temp = 0                    O(1)
for i in (0,n-1){                                        O(n)
    temp = Math.abs( S[ h( S[i] ) ] - i)
    if (temp < i){
        a = S[ h( S[i] ) ]
        b = i
    }
}
return (a,b)
Running time = O(1) + O(n) + O(1) + O(n) = O(n)
```

Problem 5 :
Before writing my pseudocode I assume that the Universal Hash Function you give us on page 6 from the lecture note for lecture 6 is an universal hash function and I can use that and I will refer that function as $h()$.

```
Declare an arraylist S of which all elements are 0 of size n  O(1)
int count = 0
for i in (0,k){                                                O(k)
    if(S[ h( A[i] ) ] != 0){
        S[ h( A[i] ) ] = 1
        count ++
    }
}
B[0] = count                                                   O(1)
for i in (1, n-1-k){                                           O(n-k)
    if(S[ h( A[i-1] ) ] == 1){
        S[ h( A[i-1] ) ] = 0
        count --
    }
    if(S[ h( A[i+k] ) ] == 0){
        S[ h( A[i+k] ) ] = 0
        count ++
    }
    B[i] = count
}
return B
```

Running time = $O(1) + O(k) + O(1) + O(n - k) = O(n)$