UM-SJTU JOINT INSTITUTE

Introduction to Algorithms
(VE477)

Homework #1

Prof. Manuel

Xinmiao Yu
518021910792

Sept. 18, 2020

Q1.

1. Obviously $1 \leq k \leq n$, otherwise the statement would not be true. To prove, because the hash table has n slots and the probability of n keys to hash to any slot is equal, for each key, it has a probability of $\frac{1}{n}$ to hash to any slot. So the number of keys hash to a same slot follows a binomial distribution with parameter n and p, where $p = \frac{1}{n}$. Then the probability for exactly k keys hash to a same plot is

$$P_k = \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \binom{n}{k}.$$

2. The probability of a slot to have k keys is P_k. More than one slots may have k keys, but no slots would have more than k keys. Then, P'_k =the probability of at least one slot has k keys and other slots have no more than k keys, which is **smaller than or equal to** the probability that at least one slot has k keys. Because we have n slots, the probability of only one slot has k keys is $\binom{n}{1} P_k = nP_k$, and this probability is **larger than or equal to** the probability that at least one slot has k keys. Through this two inequality, $P'_k \leq nP_k$.

3. We have Stirling formula $n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$, and $1 \leq k \leq n$

$$\begin{aligned}
P_k &= \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \frac{n!}{k!(n-k)!} \\
&\approx \frac{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n}{\sqrt{2\pi (n-k)} \left(\frac{n-k}{e}\right)^{n-k} k!} \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \\
&= \sqrt{\frac{n}{n-k}} \left(\frac{n}{n-k} \frac{n-1}{n}\right)^n \left(\frac{n-k}{e} \frac{1}{n} \frac{n}{n-1}\right)^k \frac{1}{\sqrt{2\pi k} \left(\frac{k}{e}\right)^k} \\
&< \sqrt{\frac{n}{2\pi k(n-k)}} \frac{e^{-k}}{\left(\frac{k}{e}\right)^k} \\
&< \frac{e^k}{k^k}
\end{aligned}$$

Q2.

Suppose G is an undirected graph G with weighted edges and the weight of an edge e is decreased where $e \notin T$, $e = (u, v)$.

Algorithm 1: Algorithms in the homework

Input : this file

Output: nice algorithms in the homework

```
Function AlgoHw(this file):
    download file;
    open file;
    compile file;
    while not at end of this document do
        read;
        if understand then
            go to next line;
            current line becomes this one;
        else if want to know more on algorithms in LaTeX then
            refer to algorithm2e documentation
        else
            restart reading from the beginning;
        end if
    end while
    for exercise ← 1 to 7 do
        if algorithm is requested then
            solve the problem;
            A[exercise] ← write the algorithm in LaTeX;
        end if
    end for
    return A
end
```

Q3.

Algorithm 2: Compute Sum

Input : two n-bits integer stored in array $num1$, $num1$ separately

Output: array $result$ stores the sum of two integers

```
Function AlgoHw(num1, num2):
    i ← 0;
    carry ← 0;
    while i < n do
        x ← num1[i] + num2[i] + carry;
        if x < 10 then
            result[i] ← x;
            carry ← 0;
        else
            result[i] ← x % 10;
            carry ← 1;
        end if
        i ← i + 1;
    end while
    if carry = 1 then
        result[i] ← 1;
    end if
    return result
end
```