CSCI 5454: Algorithms: Homework 5

Ashutosh Gandhi

March 15, 2023

Problem 1

1.1 Part A

P(a ball goes to bin j) = $1/m$
P(a ball does not go to bin j) = $1 - 1/m$
P(none of the nk balls go to bin j) = $(1 - 1/m)^{nk}$

1.2 Part B

P(one bin is unoccupied) = $(1 - 1/m)^{nk}$ # from 1.a
P(one bin is occupied) = $1 - (1 - 1/m)^{nk}$
P(k bins are occupied) = $(1 - (1 - 1/m)^{nk})^k$
using the inequality $1 - x \leq e^{-x}$

$$\begin{aligned}
(1 - 1/m) &\leq e^{-1/m} \\
(1 - 1/m)^{nk} &\leq e^{-nk/m} \\
-(1 - 1/m)^{nk} &\geq -e^{-nk/m} \\
1 - (1 - 1/m)^{nk} &\geq 1 - e^{-nk/m} \\
(1 - (1 - 1/m)^{nk}))^k &\geq (1 - e^{-nk/m})^k \quad \# \text{ since } k > 0
\end{aligned}$$

Thus, P(k bins are occupied) $\geq (1 - e^{-nk/m})^k$

1.3 Part C

Let E_1 denote probability that bin 1 is occupied, E_2 that bin 2 is occupied, $E_1 \cap E_2$ denote that bin1 and bin2 are both occupied, and $E_1 \cup E_2$ denote that either bin1 or bin2 is both occupied
Then, $P(E_1 \cap E_2) = P(E_1) + P(E_2) + P(E_1 \cup E_2))$
from 1.a; P(none of the q balls goes to a particular bin) = $(1 - 1/m)^q$
so P(E_1)) = $1 - (1 - 1/m)^q$

similarly P(E_2)) $= 1 - (1 - 1/m)^q$

For finding $E_1 \cup E_2$
P(a ball not going to bin1 or bin2) = (m-2)/m # since it could go in any of the remaining m-2 bins
P(neither bin1 or bin2 is occupied after q balls) $= (1 - 2/m)^q$
P(bin1 or bin 2 is occupied) $= 1 - (1 - 2/m)^q$
$P(E_1 \cup E_2) = 1 - (1 - 2/m)^q$

Thus, $P(E_1 \cap E_2) = 2[1 - (1 - 1/m)^q] - [1 - (1 - 2/m)^q]$
$P(E_1 \cap E_2) = 1 - 2(1 - \frac{1}{m})^q + (1 - \frac{2}{m})^q$

1.4 Part D

from 1.2 False positive rate $\approx (1 - e^{-nk/m})^k$
so 0.01 $\approx (1 - e^{-nk/m})^k$, with m=$8 * 10^6, k = 10$
Computing the above equation on WolframAlpha [1] we get n $\approx$ 797474
Thus, the number of unique elements that can be inserted is 797,474

In the case of a hashmap, the total storage would be 40*n bytes = $797474 * 40)$ Bytes or 30.42 MB
The bloom filter takes almost 1MB size, thus the overall size save is $30.42 - 0.95 \approx 29.47MB$
[1] `https://www.wolframalpha.com/input?i=%281-e%5E%28%28-n%29*%2810%2F8000000%29%29%29%5E10%3D0.01`

2.1 Part A

If a graph has a min-cut of k, then it must have at least nk/2 edges
P(choosing a min-cut in 1st step of contraction) $= k/(nk/2) = 2/n$
P(not choosing a min-cut in 1st step of contraction)$= 1 - 2/n = (n-2)/n$
Similarly, P(not choosing a min-cut in 2nd step of contraction)$= 1 - 2/n = (n-3)/(n-1)$ # from 1-(k/((n-1)*k/2))
P(preserve a min-cut from G_n to $G_{n/4}$)

$$= \frac{n-2}{n} * \frac{n-3}{n-1} * \frac{n-4}{n-2} * \frac{n-5}{n-3} * ..... * \frac{n/4}{n/4+2} * \frac{n/4-1}{n/4+1}$$
$$= \frac{n/4 * (n/4-1)}{n * (n-1)}$$
$$= \frac{n-4}{16(n-1)}$$
$$\approx 1/16 \quad \text{as n tends to infinity}$$

2.2 Part B

The time taken to contract an edge is O(n) and to do the same for n/4 edges is $O(n^2/4)$ or $O(n^2)$
Thus the overall time complexity is given by the recurrence
$T(n) = O(n^2) + 4T(n/4)$
4T(n/4) since there are 4 recursive calls being made for the n/4 contracted edge graph
Using master theorem with a=4, b=4 and c=2 we get $\log_b a < c$ Thus we use case 3 of the theorem and get T(n) = $O(n^c)$

Thus, $T(n) = O(n^2)$

2.3 Part C

Let P_n be the probability of the Recursive Algorithm succeeding then
P_n = P(1st contraction succeeds)*P(The 4 recursive calls succeeding)
P_n = P(1st contraction succeeds)*[1-$(1 - P_{n/4})^4$]

$$\begin{aligned}
P_n &= 1/16 * [1 - (1 - P_{n/4})^4] \quad \text{\# let } 4^k = \text{n} \\
p(k) &= 1/16 * [1 - (1 - p(k-1))^4] \quad \text{\# let d(k) = 1/p(k)} \\
d(k) &= \frac{16}{1 - [1 - 1/d(k-1)]^4} \\
d(k) &= \frac{16d(k-1)^4}{d(k-1)^4 - (d(k-1) - 1)^4} \quad \text{\# let d(k-1) = x} \\
d(k) &= \frac{16x^4}{x^4 - (x-1)^4} \\
&= \frac{16x^4}{4x^3 - 6x^2 + 4x - 1} \\
d(k) &= 4x + \frac{24x^3 - 16x^2 + 4x}{4x^3 - 6x^2 + 4x - 1}
\end{aligned}$$

As can be seen from Figure 1 that the graph is a decreasing function with its upper bound at x=1, since x is the inverse of probability it must be greater than or equal to 1. At x=1 the polynomial is equal to 12.

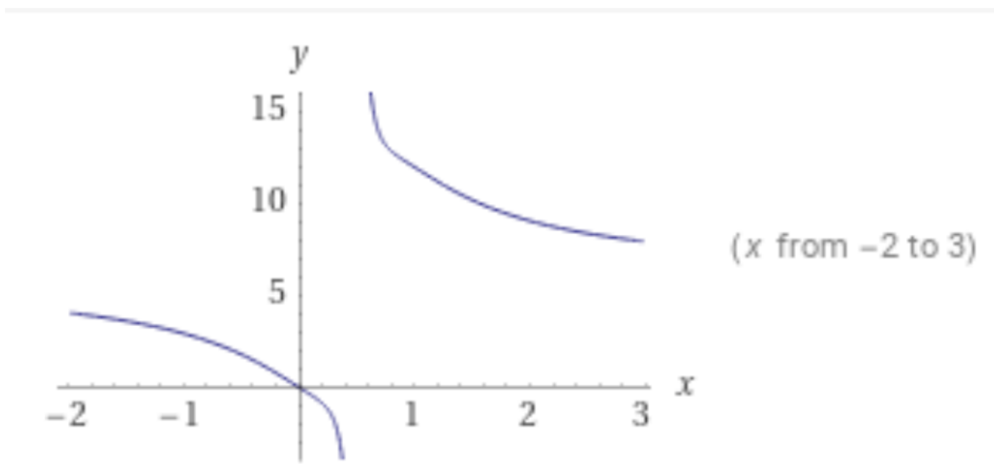

Figure 1: Graph of $\frac{24x^3-16x^2+4x}{4x^3-6x^2+4x-1}$

$$
\begin{aligned}
d(k) &\leq 4x + 12 \quad \text{substitute back x} \\
d(k) &\leq 4d(k-1) + 12 \\
&\leq 4^2 d(k-2) + [12 + 4 * 12] \\
&\leq 4^3 d(k-3) + [12 + 4 * 12 + 4^2 * 12] \quad \text{going till k-1 to get the base base d(1)=1} \\
&\leq 4^{k-1} d(1) + [12 + 4 * 12 + 4^2 * 12 + ... + 4^{k-1} * 12] \\
&\leq 4^{k-1} + 12[\frac{4^{k-1} - 1}{4 - 1}] \quad \text{apply GP sum formula with a=1, r=4, n=k-1} \\
&\leq 4^{k-1} + 4 * (4^{k-1} - 1) \\
&\leq 5 * 4^{k-1} - 4 \quad \text{substitute back p(x)} \\
p(k) &\geq \frac{1}{5 * 4^{k-1} - 4} \quad \text{substitute back n} \\
p(n) &\geq \frac{1}{5 * 4^{\log_4 n - 1} - 4} \\
&\geq \frac{1}{(5/4) * n - 4} \\
p(n) &\geq \frac{4}{5n - 16}
\end{aligned}
$$

Thus we get the lower bound $P(n) = \Omega(1/n)$

3.1 Part A

The time taken to contract an edge is O(n) and to do the same for n/2 edges is $O(n^2/2)$ or $O(n^2)$
Thus the overall time complexity is given by the recurrence
$T(n) = O(n^2) + 2T(n/2)$
2T(n/2) since there are 2 recursive calls being made for the n/2 contracted edge graph
Using master theorem with a=2, b=2 and c=2 we get $\log_b a < c$ Thus we use case 3 of the theorem and get T(n) = $O(n^c)$

Thus, $T(n) = O(n^2)$

3.2 Part B

Similar 2.a
P(preserve a min-cut from G_n to $G_{n/2}$)

$$
\begin{aligned}
&= \frac{n-2}{n} * \frac{n-3}{n-1} * \frac{n-4}{n-2} * \frac{n-5}{n-3} * ..... * \frac{n/2}{n/2+2} * \frac{n/2-1}{n/2+1} \\
&= \frac{n/2 * (n/2-1)}{n * (n-1)} \\
&= \frac{n-2}{4(n-1)} \\
&\approx 1/4 \quad \text{as n tends to infinity}
\end{aligned}
$$

Let P_n be the probability of the Recursive Algorithm succeeding then
P_n = P(1st contraction succeeds)*P(The 2 recursive calls succeeding)
P_n = P(1st contraction succeeds)*[1-$(1 - P_{n/2})^2$]

$$
\begin{aligned}
P_n &= 1/4 * [1 - (1 - P_{n/2})^2] \quad \text{\# let } 2^k = \text{n} \\
p(k) &= 1/4 * [1 - (1 - p(k-1))^2] \quad \text{\# let d(k) = 1/p(k)} \\
d(k) &= \frac{4}{1 - [1 - 1/d(k-1)]^2} \\
d(k) &= \frac{4d(k-1)^2}{d(k-1)^2 - (d(k-1) - 1)^2} \quad \text{\# let d(k-1) = x} \\
d(k) &= \frac{4x^2}{x^2 - (x-1)^2} \\
&= \frac{4x^2}{2x-1} \\
d(k) &= 2x + \frac{2x}{2x-1}
\end{aligned}
$$

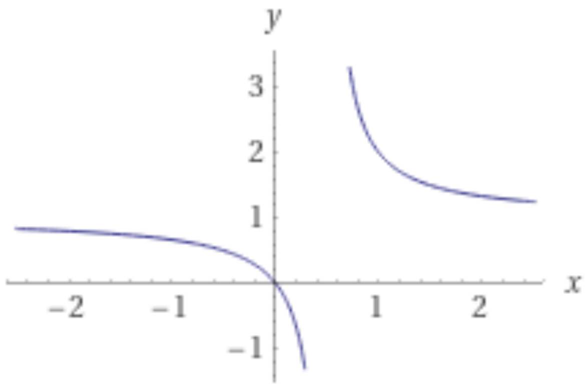

Figure 2: Graph of $\frac{2x}{2x-1}$

As can be seen from Figure 2 that the graph is a decreasing function with its upper bound at x=1, since x is the inverse of probability it must be greater than or equal to 1. At x=1 the polynomial is equal to 2.
Thus,

$$
\begin{aligned}
d(k) &\leq 2x + 2 \quad \text{substitute back d(k-1)} \\
d(k) &\leq 2d(k-1) + 2 \\
&\leq 2^2 d(k-2) + [2 + 2 * 2] \\
&\leq 2^3 d(k-3) + [2 + 2 * 2 + 2^2 * 2] \quad \text{going till k-1 to get the base d(1)=1} \\
&\leq 2^{k-1} d(1) + 2 * [1 + 2 * 1 + 2^2 * 1 + ... + 2^{k-1} * 1] \\
&\leq 2^{k-1} + 2 * \frac{2^{k-1} - 1}{2 - 1} \quad \text{apply GP sum formula with a=1, r=2, n=k-1} \\
d(k) &\leq 3 * 2^{k-1} - 2 \quad \text{substitute back p(x)} \\
p(n) &\geq \frac{1}{3 * 2^{k-1} - 2} \quad \text{substitute back n} \\
p(n) &\geq \frac{1}{3 * 2^{log_2 2 - 1} - 2} \\
p(n) &\geq \frac{1}{(3/2) * n - 2} \\
p(n) &\geq \frac{2}{3n - 4}
\end{aligned}
$$

Thus we get the lower bound $P(n) = \Omega(1/n)$