Exercises

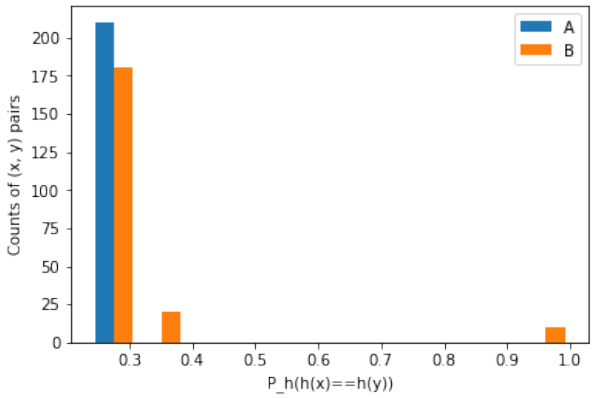
1.

$$\begin{aligned} P_h\{h(x) = h(y)\} &= \sum_{k=0}^{n-1} P(h(x) = k \text{ and } h(y) = k) \\ &= \sum P(h(x) = k)P(h(y) = k) \text{ (independence)} \\ &= \sum \frac{1}{n^2} = \frac{1}{n} \end{aligned}$$

2. A is a universal hash family. B is not.

   The following script checks if $P_h\{h(x) = h(y)\} = \frac{1}{n}$ is true for each pair $x \neq y$, for both A and B. The resulting plot (histograms of the cumulative counts of (x, y) pairs vs. P) is included.

```
def findP(A):
  outA = []
  for x in range(20):
    for y in range(x+1, 21):
      tot, cnt = 0, 0
      for h in A:
        tot += 1
        if h(x)==h(y):
          cnt += 1
      outA.append(cnt/tot)
  return outA
outA = findP(A)
outB = findP(B)
plt.hist([outA, outB], label=["A", "B"])
plt.xlabel("P_h(h(x)==h(y))")
plt.ylabel("Counts of (x, y) pairs")
plt.legend()
plt.show()
```



Problems

---

1. (a) We need $P_h \leq 1/n$ but

$$P_h(h(x) == h(y)) = \begin{cases} 0, & \text{if } x \neq y \bmod n. \\ 1, & \text{otherwise.} \end{cases}$$

   (b) Yes it is true that $P_{x,y}(h(x) = h(y)) \leq 1/n$ but $P_{x,y}$ is irrelevant in the discussion of universal hash families. To calculate $P_{x,y}$, we first list all possible values of x and y:

$$\begin{aligned} &\{0n + x | x = 0, ..., n-1\} \cup \\ &\{1n + x | x = 0, ..., n-1\} \cup \\ &\{2n + x | x = 0, ..., n-1\} \cup \\ &... \\ &\{(n-1)n + x | x = 0, ..., n-1\} \end{aligned}$$

   We then can calculate:

$$P_{x,y}(h(x) = h(y)) = P_{x,y}(x = y \bmod n) = \frac{n(n-1) \cdot n}{n^2(n^2-1)} = \frac{1}{n+1} \leq \frac{1}{n}$$

   Again this result does not guarantee that h belongs to a universal hash family.

   (c) No, as explained above.

2. (a) Counter-example: Generate a rb tree by inserting in the following order: 5, 4, 7, 6, 8. The resulting tree is below. The label and the key of each node are labeled. We have |T|=5, $|T_L|$=1, $|T_R|$=3.

```
    b,5
    /  \
 b,4   b,7
       /   \
     r,6   r,8
```

   (b) We define

   - d(x): the number of descendants of x, not counting NILs, including x.
   - b(x): the "black height". I.e. the number of black nodes from x to any NIL, excluding x.
   - h(x): the height. I.e. the number of nodes from x to the farthest NIL, excluding x.

   We have

$$2b(x) \geq h(x) \geq b(x) \tag{1}$$

$$2^{2b(x)} - 1 \geq 2^{h(x)} - 1 \geq d(x) \geq 2^{b(x)} - 1 \geq 2^{h(x)/2} - 1 \tag{2}$$

   Eq. 1 is a property of the red-black tree. The left half of Eq. 2 is the property of a "full" binary tree. The right half is due to the red-black tree.

   Depending on the color of y, we have

$$b(y) = \begin{cases} b(x), & \text{if y is red.} \\ b(x) - 1, & \text{otherwise.} \end{cases}$$

Therefore, we have

$$|T_L| = d(y) \geq 2^{b(y)} - 1 \geq 2^{b(x)-1} - 1 = \frac{2^{b(x)}}{2} - 1$$
$$= \frac{\sqrt{2^{2b(x)}}}{2} - 1 > \frac{\sqrt{2^{2b(x)} - 1}}{2} - 1 \geq \frac{\sqrt{d(x)}}{2} - 1 = \frac{\sqrt{|T|}}{2} - 1$$

The same applies to T_R.

3. (a) Suppose we know m and therefore n=10m in advance. We check if

$$P(\text{No collision for each color i}) \geq \frac{9}{10}$$

is true over both the choice of h and of the geese.

i. Over the choice of geese: Since the required inequality only concerns one color, we have

$$P = \prod_{k=1}^{m-1}(1 - \frac{1}{n}) = (1 - \frac{1}{n})^{m-1} = (1 - \frac{1}{10m})^m(1 - \frac{1}{10m})^{-1}$$

We require $P \geq .9$. Since $P \xrightarrow{\infty} e^{-.1} > .9$ and $P \geq .9$ for all $m \geq 1$. The requirement is always met. If the color of concern is not seen in the flock, we drop the $()^{-1}$ term in P; the analysis remains the same.

ii. Over the choice of h:

$$\begin{aligned}
&P(\text{No collision for color i}) \\
&= 1 - P(\text{At least one collision on color i}) \\
&= 1 - P(\bigcup_{\substack{x=0 \\ x \neq i}}^{m-1} \text{x collides with i}) \\
&\geq 1 - \sum_{\substack{x=0 \\ x \neq i}}^{m-1} P(h(x) = h(i)) \\
&= 1 - \frac{m-1}{n} = \frac{9m-1}{10m} \geq \frac{9}{10}
\end{aligned}$$

The first inequality uses the property of universal hash families and the union bound. If color i is not seen in the flock, the $m-1$ term in the last line become m; the analysis remains the same.

Therefore, using a hash table with one stored hash function and $n = 10m$ bins satisfies the criteria in probability, space, and time complexity.

(b) Option 1 (doesn't work):
One hashing function with n=100m bins. Following the same reasoning as in (a), we get

$$(1 - \frac{1}{100m})^{m-1} = (1 - \frac{1}{100m})^m(1 - \frac{1}{100m})^{-1} \geq 1 - \frac{1}{10^{10}}$$

Since the LHS (1) is ~ 0.991 at m=2, and (2) approaches $e^{-1/100} \sim 0.990$ from above, the inequality above cannot be satisfied (other than in the trivial case m=1) so we cannot use one hash function with n=100m bins.
Option 2:

Two arrays, one for recording numbers and the other for names of colors. We are provisioned 100m numbers, which typically means 400m bytes. The first array is of length 4m bytes. It records the numbers of geese seen in each color. The other array is of length 396m bytes, allowing the name of each color to be 395 bytes. As long as every color can be identified by a string shorter than 395 bytes, this method satisfies all criteria.

Option 3:

We choose 10 hash functions independently from the universal family. We use 10 arrays, each of length $n = 10m$, as hash tables. Therefore, the space requirement is met.

The update and the query operations are both O(1). For each entry, we hash it 10 times and save each result in the corresponding array. That is,

```
Array[i, h[i](x)] += 1
```

At query, we output the minimum of all arrays. That is,

```
output = min([Array[i, h[i](x)]] for i in len(Array))
```

Probability of correctness:

From (a), we have $P(\text{One h encounters no collision for color i}) \geq .9$. Here, we only need one of the h's to encounter no collision for color i, in order for the query to be correct. Therefore,

$$
\begin{aligned}
&P(\text{At least one h encounters no collision}) \\
&= 1 - P(\text{Every h encounters collision}) \\
&= 1 - P^{10}(\text{One h encounters collision}) \\
&= 1 - (1 - P(\text{One h encounters no collision}))^{10} \\
&\geq 1 - \frac{1}{10^{10}}
\end{aligned}
$$