CISC 471 HW 1 Part 2

Hershil Devnani (20001045)

January 22, 2021

1 Programming

Please refer to the submitted python files and the README.md for the solutions to Part 1, 2.2 and the pseudoecode for 2.3.

2 Theory

2.1 Indices

Please refer to the **indices.py** file for the coded implementation of the algorithm.

Question 1: Suppose all of the values n_1, n_2, . . . , $n_d = N$. If we say an output of $(0, 0, ..., 0)$ is one unit of output, how many units of output in terms of N and d are there?

Assuming all values of n_1, n_2, . . . , $n_d = N$, the number of units of output, in terms of N and d, will be:

$$(N + 1)^d$$

Question 2: Now dropping the assumption that n_1, n_2, . . . , $n_d = N$, how many units of output in terms of n_1, n_2, . . . , $n_d = N$ and d are there?

Assuming all values of n_1, n_2, . . . , $n_d \, ! = N$, the number of units of output will be:

$$1, 2, 4, 6, 9, 12, 16, 20, 25, 30, 36$$

Note that the number of units of output is in order from having, for example, $d = 3$, and the values of N going from $(0, 0, 0), (0, 0, 1), (0, 1, 1)$ ... (n_1, n_2, n_3). This is the same for any value of d and any values for n_i.

In terms of N and d, an equation to model this can be:

$$\begin{cases} \dfrac{(\text{sum of } n's + 2)^2}{4} & \text{if sum of } n's \text{ is even} \\ \dfrac{(\text{sum of } n's + 2)^2 - 1}{4} & \text{if sum of } n's \text{ is odd} \end{cases}$$

or, an equivalent and slightly simplified version that I found afterwards was

$$(n_1 + 1)(n_2 + 1)...(n_d + 1)$$

2.2 Loops

Prove that

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

Proof.

- We will prove by induction that, for all $n \in \mathbb{N}, n \geq 1$,

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \tag{1}$$

- *Base Case:* When $n = 1$.

  When $n = 1$, the left side 1, and the right side is

$$\frac{1(1+1)}{2} = \frac{2}{2} = 1$$

  So the both sides are equal and this equation, equation 1, holds true.

- *Induction Step:* When $n = 1$.

  Let $k \in \mathbb{N}$ be given and suppose equation 1 is true for $n = k$. Then,

$$
\begin{aligned}
\sum_{i=1}^{k+1} i &= \frac{n(n+1)}{2} \\
&= \sum_{i=1}^{k+1} i + (k+1) \qquad \textit{(by induction hypothesis)} \\
&= \frac{k(k+1)}{2} + (k+1) \\
&= (k+1)(\frac{k}{2} + 1) \\
&= (k+1)(\frac{k+2}{2}) \\
&= \frac{(k+1)(k+2)}{2}
\end{aligned}
$$

- *Conclusion:* By the principle of induction, equation 1 is true for all $n \in \mathbb{N}, n \geq 1$.

Thus, equation 1 holds for $n = k + 1$ and the proof of the induction step is complete.

□

2.3 Dishonest Professors

Pseudocode is also provided in **dishonest_profs_pseudocode.py**.

What we know:

- n = 100 professors
- Number of dishonest professors, ℓ, is $1 \leq \ell <$ n/2

Example (General Idea):

- Let's say we have 10 professors.
- Now we begin to ask all the other professors if the selected professor is honest or dishonest.
- Once we reach a point where more people have said the professor is dishonest, then we stop for a second.
- Say 3 people said the selected professor was honest and 4 said he was dishonest.
- Including the dishonest professor, we have 8 people, and we know half of them are lying.

- We know we have 4 liars, so we can reset out threshold for the remaining number of liars to ℓ - 4.
- We remove these 8 professors from consideration and select a new professor.
- We repeat this until we get an ℓ number of professors classifying the selected professor as now we know we have passed the threshold for the number of dishonest professors.

Explanation:

The following explains a process to attempt to find the set of honest and dishonest professors out of 100, with only being able to ask a maximum of 198 questions.

There are more honest professors than dishonest professors. So at most, there are 49 dishonest and 51 honest professors. Next, we begin asking the remaining professors one-by-one about whether the selected professor is honest or dishonest.

As soon as we reach a case in which more people have said the the selected professor is a dishonest professor, then we know that out of the x number of people that have accused the selected professor of being dishonest, x - 1 professors have supported the selected professor being honest.

Out of the $2x$ people involved up to the point, being the selected professor and the professors that have been asked a question so far. Out of the $2x$ asked, at least x are dishonest professors. Since if say even one professor says the selected professor is dishonest, assuming a dishonest professor will not admit he is dishonest, one of the two professors is lying.

We can ignore these $2x$ professors and replace our threshold for ℓ to ℓ - x, meaning that at most we have ℓ - x remaining dishonest professors.

We repeat this process until we reach a state where ℓ professors have classified the selected professor as an honest professor. At this point we know that we have found an honest professor since the threshold of the maximum possible number of dishonest professors has been succeeded, so the majority vote has to be the correct vote. Therefore we have found an honest professor.

It is enough to stop when ℓ professors have identified the selected professor as honest, and not $\ell + 1$, is that we don't ask the selected professor themselves, i.e. if the selected professor is dishonest, we have at most ℓ - 1 remaining dishonest professors, so as long as we get ℓ professors to testify the selected professor is an honest professor, we know the selected professor is honest.

At most, this process would have taken n + ℓ - 1 questions, so for 100 professors, this would have taken 100 - 49 - 1 = 50 question so far.

Now, we can ask this honest professor about all the other professors. This adds another 99 questions that we ask, so in total we have asked 149 questions, and within the requirements of 198 questions we have found out all honest and dishonest professors amongst the bioinformatics department at this prominent university.