Induction

Peter Poon

September 29, 2013
Show that $1 + \frac{1}{4} + \frac{1}{9} + \ldots + \frac{1}{n^2} \leq 2 - \frac{1}{n}$
Show that \( 1 + \frac{1}{4} + \frac{1}{9} + \ldots + \frac{1}{n^2} \leq 2 - \frac{1}{n} \)

Base Case:
When \( n = 1 \), \( 1 \leq 2 - \frac{1}{n} \), so it is true.

Induction step:
Assume it is true when \( n = k \),
when \( n = k + 1 \)

\[
1 + \frac{1}{4} + \frac{1}{9} + \ldots + \frac{1}{k^2} + \frac{1}{(k+1)^2} \leq 2 - \frac{1}{k} + \frac{1}{(k+1)^2} \text{ by assumption}
\]

\[
= 2 - \frac{(k+1)^2 - k}{k(k+1)^2}
\]
You want to show that \(2 - \frac{(n+1)^2 - n}{n(n+1)^2} \leq 2 - \frac{1}{n+1}\) for positive integer \(n\).

\[
2 - \frac{(n + 1)^2 - n}{n(n+1)^2} \leq 2 - \frac{1}{n+1}
\]

\[
\frac{(n + 1)^2 - n}{n(n+1)^2} \geq \frac{1}{n+1}
\]

\[
\frac{(n + 1)^2 - n}{n(n+1)} \geq 1
\]

\[
(n + 1)^2 - n \geq n(n + 1)
\]

\[
n^2 + n + 1 \geq n^2 + n
\]

\[
1 \geq 0
\]

So, \(1 + \frac{1}{4} + \frac{1}{9} + \ldots + \frac{1}{(k+1)^2} \leq 2 - \frac{(k + 1)^2 - k}{k(k+1)^2} \leq 2 - \frac{1}{k+1}\).
You know how to construct binary Gray code.

How about a $n$ digit Gray code that can use $\{0, 1, 2\}$ for each digit.
Show that there exist some ways to construct $n$ digits gray code if each digit can be 0, 1, or 2.

For example when $n = 2$, the Gray code can be $\{00, 01, 02, 12, 10, 11, 21, 22, 20\}$. 

When \( n = 1 \), one possible solution is \( \{1, 2, 0\} \)

Assume you can construct a \( k \) digits Gray code.

Assume your \( k \) digits code is \( \{x_1, x_2, x_3, \ldots, x_{3^k - 1} x_{3^k}\} \)
If you reverse the order, you will find another \( k \) digits code.
\( \{x_{3^k}, x_{3^k - 1}, x_{3^k - 2}, \ldots, x_2, x_1\} \)

e.g. let \( k = 2 \), \( x_1 = 00 \), \( x_2 = 01 \), \( x_3 = 02 \), \ldots, \( x_9 = 20 \)
The original code is \( \{00, 01, 02, 12, 10, 11, 21, 22, 20\} \).
The reverse order is \( \{20, 22, 21, 11, 10, 12, 02, 01, 00\} \).
Then this is a valid construction for $k + 1$ digits.

The red part indicates where the only different digit at

$0x_1$

$0x_2$

... 

$0x_{3^k}$

$1x_{3^k}$

$1x_{3^k-1}$

... 

$1x_1$

$2x_1$

$2x_2$

... 

$2x_{3^k}$


e.g. when there are 3 digits, 000

001

002

... 

022

020

120

122

121

... 

101

100

200

201

... 

220
There are $n$ people in CUHK, they want to form some organizations. People can join more than one organization. However, there are some rules.

1. Every organization must have at least one person.

2. No two organizations consist of the same group of people.

3. If some people are in both organizations A and B, then either every person in A are in B or every person in B are in A.

Show that the number of organizations will not exceed $2n - 1$. 
When $n = 1$, you can only form one organization.

Assume it is true for all $i, 1 \leq i \leq k$,

When there are $k + 1$ people, there must be some organization $A$ such that no other organization $B$ contains all people in $A$.

Let’s call such organizations the top organization. Assume there are $h$ top organizations, and each organization $i$ have $P_i$ people.

e.g. The top organizations are A, B and C.
Case 1: Every $P_i < k + 1$.

The top organization are A1, A2, ..., Ah.

Then total number of organizations

\[ = \text{number of organizations form by people in A1} \]
\[ + \text{number of organizations form by people in A2} + \ldots \]
\[ + \text{number of organizations form by people in Ah} \]
\[ \leq 2P_1 - 1 + 2P_2 - 1 + \ldots + 2P_h - 1 \text{ By Induction hypothesis} \]
\[ = 2 \sum_{i=1}^{h} P_i - \sum_{i=1}^{h} 1 \]
\[ \leq 2(k + 1) - h \text{ sum of } P_i \text{ can’t exceed } k + 1 \]
\[ \leq 2(k + 1) - 1 \]
What if there is one large organization that contain every people? Then we can’t apply assumption directly.

Let A be that organization. Then if we remove A, there will be $m$ top organizations inside A, lets call them $B_i$, $1 \leq i \leq m$.

There will be two cases.

Case 2a: $m = 1$ and $B_1$ contains at most $k$ people.

By assumption, total number of organization

$= 1 + \text{number of organization formed in } B_1$

$\leq 1 + 2k - 1 = 2(k + 1) - 2.$
Case 2b: There are $m > 1$ top organizations and each have $Q_i$ people.

Total number of organizations

\[
= 1 + \text{number of organization formed in B1} \\
+ \text{number of organization formed in B2} + \ldots \\
+ \text{number of organization formed in Bm} \\
\leq 2Q_1 - 1 + 2Q_2 - 1 + \ldots + 2Q_m - 1 + 1 \\
= 2 \sum_{i=1}^{m} Q_i - m + 1 \\
\leq 2(k + 1) - m + 1 \\
\leq 2(k + 1) - 1 \text{ because } m > 1
Problem 4

Ceiling of $x = \text{the smallest integer } y \text{ such that } y \geq x$.

ceiling example:
ceiling of $0.7 = 1$
ceiling of $2 = 2$
ceiling of $-2 = -2$
ceiling of $-0.5 = 0$
Goal of the algorithm
Input: \( \frac{a}{b} \)
Output: find \( \{x_1, x_2, \ldots, x_k\} \) such that \( \frac{1}{x_1} + \frac{1}{x_2} + \ldots + \frac{1}{x_k} = \frac{a}{b} \)

Example of algorithm
Input: \( \frac{7}{10} \)
\[ \lceil \frac{10}{7} \rceil = 2, \quad \frac{7}{10} - \frac{1}{2} = \frac{2}{10} \]
\[ \lceil \frac{10}{2} \rceil = 5, \quad \frac{2}{10} - \frac{1}{5} = 0 \]
So, \( \frac{7}{10} = \frac{1}{2} + \frac{1}{5} \)
Peter’s idea

Let input fraction be $f$.

After 1 iteration of algorithm, I will have a new fraction $f'$ and $f' < f$

The problem becomes finding solution for input $f'$.

If my induction hypothesis is the algorithm finishes in finite number of step when input fraction $< f$.

When the input is $f'$, the algorithm will stop by assumption. So $f$ will finishes in finite number of step.
Problem 3
You want to design a hardware that take $n$ integers as input, and output them in ascending order.

E.g.

![Diagram showing a design that outputs numbers in ascending order]
You can add some comparators to network.

A comparator takes 2 input, and output them in ascending order. So it is a sorting network of size 2.
Incorrect example of sorting network of size 3

Correct example of sorting network of size 3