演算法 Homework 20191212
山下夏輝(Yamashita Natsuki)
R08922160

# Mini HW #8

課題の提出

**期限** 木曜日 14.20 まで　　**点数** 1
**提出しています** テキスト入力ボックスまたはファイルのアップロード　　**ファイル タイプ** pdf
**使用可能** 12月12日 日 14.20 ～ 12月19日 日 14.20 7 日

Consider a modified n-bit counter, let $a_1$ be its lowest bit, $a_n$ be its highest bit, and all the bits be 0 initially. When it counts to $X$, it will flip all $a_i$ where $i$ is a multiple of $X$. Hence, the counter can only count from 0 to $n$.

Each flip on a bit will incur a 1-unit cost. For example, when a 4-bit counter reads "1111", then its value will be 1, and when it counts to 2, $a_2$ and $a_4$ will be flipped and the counter therefore reads "0101" after the counting operation which costs 2 units.

(1) (20%) How does an 7-bit counter read after it counts from 0 to 5, and what is the total cost of the five counting operations?
(2) (80%) What is the amortized cost for one counting operation on an n-bit counter counting from 0 to $n$? Please select one of the methods mentioned in class (aggregate/accounting/potential) to show how you derive the answer.
(3) (Bonus 10%) Prove that the reading of the counter will be always unique when counting from 0 to $n$

(1)

| Cnt | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Cost | Total |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | **1** | **1** | **1** | **1** | **1** | **1** | **1** | 7 | 7 |
| 2 | 1 | **0** | 1 | **0** | 1 | **0** | 1 | 3 | 10 |
| 3 | 1 | **1** | 1 | 0 | **0** | 0 | 1 | 2 | 12 |
| 4 | 1 | 1 | 1 | **1** | 0 | 0 | 1 | 1 | 13 |
| 5 | 1 | 1 | **0** | 1 | 0 | 0 | 1 | 1 | 14 |

How to count is indicated above.
Total cost of the five counting operation is 14.

(2)
With accounting method
2 for setting a bit to 1, because initial setting is all 0 and the first time when costs come up is time to count 1 from 00⋯ to 11⋯. Then,

Bit 0 -> bit 1　amortized cost +2
Bit 1 -> bit 0　amortized cost 0

Each bit 0 to 1, we add 2 to the amortize costs, which the costs of each bit 1 flipped to 0 are saved in advance. Then, flipping each bit 0 to 1, the saved cost in advance can be used. (So when you flip each bit 0 to 1, you do not need to save any cost.)

In addition to it, let we think the condition below. This is the case with the largest cost because every bits is flipped in each step. The total cost is 8. The cost in each step is 4. In this way to calculate the amortized cost, when each bit 0 to 1, 8 amortized cost is saved. It is the enough costs to cover the cost that every 1 is flipped to 0. Therefore, this way to calculate the amortized cost is correct.
init: 0000
step1: 1111
step2: 0000.

The demonstration of the calculation of the amortized cost with the example is below.

| Cnt | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Cost | Total | Amo | Amo-Total |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | **1** | **1** | **1** | **1** | **1** | **1** | **1** | 7 | 7 | 14 | 14 |
| 2 | 1 | **0** | 1 | **0** | 1 | **0** | 1 | 3 | 10 | 0 | 14 |
| 3 | 1 | **1** | 1 | 0 | **0** | 0 | 1 | 2 | 12 | 2 | 16 |
| 4 | 1 | 1 | 1 | **1** | 0 | 0 | 1 | 1 | 13 | 2 | 18 |
| 5 | 1 | 1 | **0** | 1 | 0 | 0 | 1 | 1 | 14 | 0 | 18 |