CSCE 411: Design and Analysis of Algorithms

Lecture 8: Amortized Analysis

Date: February 11, 2025 *Nate Veldt, updated by Samson Zhou*

Course Logistics

- Amortized analysis: Chapter 17
- First test is Thursday

1 Representing numbers in binary

We represent a number $x \in \mathbb{N}$ in binary by a vector of bits $A[0..k]$, where $A[i] \in \{0, 1\}$ and

$$x = \sum_{i=0}^{k} A[i] \cdot 2^i$$

2 The Binary Counter Problem

Let INCREMENT be an algorithm that takes in a binary vector and adds one to the binary number that it represents.

INCREMENT(A)
  $i = 0$
  while $i < A.length$ and $A[i] == 1$ **do**
    $A[i] = 0$
    $i = i + 1$
  end while
  if $i < A.length$ **then**

    ____

  end if

Question 1. If A is length k binary vector, what is the worst-case runtime for calling INCREMENT(A)?

A $O(1)$

B $O(\log k)$

C $O(k)$

D $O(n)$

2.1 The actual cost of each iteration

Assume it takes $O(1)$ time to check an entry of A or to flip its bit. At each step, the runtime is then just $O(\text{number of flipped bits})$, so we will say the cost of an iteration is

$c_i =$ ____

Key idea: Separate the costs that happen ______________________

0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1
10	0	1	0	1	0
11	0	1	0	1	1

2.2 Technique 1: Aggregate Analysis

Let $T(n)$ be the total number of number of flipped bits in n increments.

Question 2. Look at the table on the last page, and observe how often the bit in position $A[j]$ is flipping. If you wish, fill in the number of total flipped bits at each increment. What is the total cost of calling Increment n times?

- A $O(n)$
- B $O(k)$
- C $O(nk)$
- D $O(n^2)$

2.3 Technique 2: The accounting method

- c_i = the actual number of bits flipped
- $\hat{c}_i$ amortized cost for flipping bits: ________________________

0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1

```
INCREMENT(A)
  i = 0
  while i < A.length and A[i] == 1 do
      A[i] = 0
      i = i + 1
  end while
  if i < A.length then
      A[i] = 1
  end if
```

2.4 Technique 3: The potential method

Step 0: Identify data structure, potential function, and costs.

- c_i: the actual number of flipped bits at iteration i
- data structure: __________________________, with state D_i after iteration i
- $\Phi(D_i) = b_i =$ ____
- $\hat{c}_i =$ ______________________________________

Step 1: Check that $\Phi(D_i) \geq \Phi(D_0)$

Step 2: Compute and bound amortized costs

Let t_i be the number of bits that are set to 0 (*while* loop in algorithm) in iteration i.

If $b_i = 0$, this means we had $A = [1\ 1\ 1\ \cdots\ 1]$ in iteration i, and incrementing turned it into $A = [0\ 0\ 0\ \cdots\ 0]$, meaning $b_{i-1} = t_i = k$.

Question 3. *If $b_i > 0$, which of these is the right expression for b_i?*

A $b_i = b_{i-1} + 1$

B $b_i = t_i + 1$

C $b_i = b_{i-1} + t_i$

D $b_i = b_{i-1} - t_i$

E $b_i = b_{i-1} + t_i + 1$

F $b_i = b_{i-1} - t_i + 1$

Whether or not $b_i = 0$, we have a bound of ________________________________

We can then compute the amortized cost and overall runtime bound: