Design and Analysis of Algorithms

6.3 Dynamic Programming Exercises

Daniel Shannon

May 18th, 2022

7.4.2

Suppose you have a binary array.

You are using it as a counter:

- In each increment, increase value by 1.
- e.g. $[0, 0, 1, 1, 1, 0, 1] \rightarrow [0, 0, 1, 1, 1, 1, 0]$

Suppose that it costs \$1 to flip a bit.

Keep a "bank" for each bit.

Suppose whenever you flip a bit from $0 \rightarrow 1$, you pay \$2: \$1 to flip, \$1 to that bit's bank.

Perform amortized runing-time analysis.

Start at right. If 0, flip to 1, terminate. If 1, flip to 0, move left and repeat.

1. How many 0→1 flips per increment?

   1 flip per increment

2. What is the cost per increment?

   \$2 per increment

3. For a 1→1 flip, how much money is in the bit's bank?

   \$1 will be in the bit's bank for a 1→0 flip.

Amoratized cost \$2 $O(n)$

7.4.4

Atomic ops=individual flips

- Over n increments, how often do we flip the first bit?

  2^n times

  Answer: **n**

- How often do we flip the second bit?

  2^{n-1}

  Answer: $\frac{n}{2}$

- How often do we flip the k^{th} bit?

  2^{n-k}

  Answer: $\frac{n}{2^{k-1}}$

- What is the total running time?

  $log(n)$

  Answer: $n + \frac{n}{2} + \frac{n}{4} + ... \leq 2n$ $O(n)$

7.4.6

- Suppose you have a different binary counter.
- But now the cost to flip the k^{th} bit is 2^k (k starts at 0).
- What is the total running time?
  - Over n increments, how often do we flip the first bit?
    2^n times
    Answer: **n**
  - How often do we flip the second bit?
    $2^{n/2}$
    Answer: **n/2**
  - How often do we flip the k^{th} bit?
    2^{n-k}
  - What is the total running time?
    $log(n)$
    Answer: $n + 2\frac{n}{2} + 4\frac{n}{4} + ... = n + n + n... = O(nlog(n))$