Amortized Analysis

Definition
The **amortized cost** of a sequence of operations is the total cost divided by the number of operations.

- Same as average-case?

- Compare to worst-case and best-case.

- Mainly a data structures issue?

Binary Counter

Problem
Increment a bit-string from an integer k to $k + 1$

- Best-case cost:

- Worst-case cost:

Amortized version: Consider n **consecutive** increments.

Amortized Binary Counter

Counter	Update cost
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	

How often does 1 bit change?

How often do 2 bits change?

How often do 3 bits change?

. . .

Total cost for n increments:

Amortized cost for *each* increment:

Self-Organizing Search

Problem
Keep reordering an array to optimize a sequence of **linear searches**.

- Supports what ADT?

- Parameters: n distinct elements, $m > n$ searches

- Request frequencies: $k_1 \geq k_2 \geq \cdots k_n$

- How to analyze an approach?

Approaches

1. **Optimal Static Ordering** (OPT)
   Look at the **entire request sequence** and determine frequencies.
   Order elements by decreasing frequency.

2. **Move-to-front** (MTF)
   Move each element to the front **after it is accessed**.

3. **Transpose**
   Swap each element with its predecessor after it is accessed.

Which way is best?

Competitive Analysis

Idea: Don't analyze worst-case, compare to optimal!

- Transpose:

- MTF: