University of Ljubljana, Faculty of Computer and Information Science

Amortized analysis of computational complexity

Prof Marko Robnik-Šikonja

Analysis of Algorithms and Heuristic Problem Solving
Edition 2024

Three methods

- aggregated analysis
- the accounting method
- the potential method

- Cormen et al, Chapter 17

Aggregated analysis

- Two examples:
  - stack
  - binary counter

Example1: Stack with MULTIPOP operation

```
MULTIPOP(S, k)
    while not STACK-EMPTY(S) and k > 0
        POP(S)
        k = k - 1
```

Example 2: Incrementing binary counter

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

Binary counter: aggregated analysis

Example: $k = 3$

[Underlined bits flip. Show costs later.]

counter value	A 2 1 0	cost
0	0 0 <u>0</u>	0
1	0 <u>0 1</u>	1
2	0 1 <u>0</u>	3
3	<u>0 1 1</u>	4
4	1 0 <u>0</u>	7
5	1 <u>0 1</u>	8
6	1 1 <u>0</u>	10
7	<u>1 1 1</u>	11
0	0 0 <u>0</u>	14
⋮	⋮	15

Cost of INCREMENT $= \Theta(\text{\# of bits flipped})$.

Binary counter: aggregated analysis

Not every bit flips every time.

[Show costs from above.]

bit	flips how often	times in n INCREMENTS
0	every time	n
1	1/2 the time	$\lfloor n/2 \rfloor$
2	1/4 the time	$\lfloor n/4 \rfloor$
	⋮	
i	$1/2^i$ the time	$\lfloor n/2^i \rfloor$
	⋮	
$i \geq k$	never	0

Stack: accounting method

Stack

operation	actual cost	amortized cost
PUSH	1	2
POP	1	0
MULTIPOP	$\min(k, s)$	0

Stack: potential method

operation	actual cost	$\Delta\Phi$	amortized cost
PUSH	1	$(s+1)-s=1$	$1+1=2$
		where $s = \#$ of objects initially	
POP	1	$(s-1)-s=-1$	$1-1=0$
MULTIPOP	$k' = \min(k, s)$	$(s-k')-s=-k'$	$k'-k'=0$

Example 3: Dynamic arrays

```
TABLE-INSERT(T, x)
if T.size == 0
    allocate T.table with 1 slot
    T.size = 1
if T.num == T.size
    allocate new-table with 2 · T.size slots
    insert all items in T.table into new-table
    free T.table
    T.table = new-table
    T.size = 2 · T.size
insert x into T.table
T.num = T.num + 1
```