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
March 2022

Three methods

- aggregated analysis
- the accounting method
- the potential method

- CLRS, Chapter 17

Aggregated analysis

- Two examples:
  - stack
  - binary counter

Example1: Stack with MULTIPOP operation

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

Example 2: Incrementing binary counter

```
INCREMENT(A)
1   i = 0
2   while i < A.length and A[i] == 1
3       A[i] = 0
4       i = i + 1
5   if i < A.length
6       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 $\underline{0}$	0
1	0 $\underline{0\ 1}$	1
2	0 1 $\underline{0}$	3
3	$\underline{0\ 1\ 1}$	4
4	1 0 $\underline{0}$	7
5	1 $\underline{0\ 1}$	8
6	1 1 $\underline{0}$	10
7	$\underline{1\ 1\ 1}$	11
0	0 0 $\underline{0}$	14
⋮	⋮	15

Cost of INCREMENT $= \Theta($# 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	$\lfloor n/4 \rfloor$
	$\vdots$	
i	$1/2^i$ the time	$\lfloor n/2^i \rfloor$
	$\vdots$	
$i \geq k$	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	$(s+1)-s=1$ where $s=$ # of objects initially	$1+1=2$
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)
 1  if T.size == 0
 2      allocate T.table with 1 slot
 3      T.size = 1
 4  if T.num == T.size
 5      allocate new-table with 2 · T.size slots
 6      insert all items in T.table into new-table
 7      free T.table
 8      T.table = new-table
 9      T.size = 2 · T.size
10  insert x into T.table
11  T.num = T.num + 1
```

Exercise: Cuckoo hashing

- See the paper

Rasmus Pagh (2006): Cuckoo Hashing for Undergraduates.