Analysis and Design of Algorithms

NUS
National University of Singapore

Algorithms
CS3230

Tutorial

Week 9

Question 1

Which of the following statements is **false**?

- The amortized cost for insert in dynamic tables is $\Theta(1)$.
- In the accounting method, the amortized cost $\hat{c}_i$ is always greater than the actual cost c_i of an operation.
- $\sum_{i=1}^{n} \hat{c}_i - \sum_{i=1}^{n} c_i \geq 0$ where $\hat{c}_i$ and c_i are the amortized and actual costs of the i-th operation respectively.

Question 2

- Consider a data structure that is based on a queue with four operations:
  - ENQUEUE(a): Add the element a into the queue
  - DEQUEUE(): Dequeue a single element from the queue
  - DELETE(k): Dequeue k elements from the queue
  - ADD(A): Enqueue all elements in A
- **Claim**: ENQUEUE, DEQUEUE and DELETE run in amortized O(1) time while ADD runs in amortized O(|A|) time.
- Using accounting method, can you show that these time complexities are correct?
- (Please state the charge for each operation.)

Potential Method (Recap)

ϕ: Potential function associated with the algorithm/data-structure

$\phi(i)$: Potential at the end of ith operation

Important conditions to be fulfilled by ϕ

$\phi(0) = 0$

$\phi(i) \geq 0$ for all i

Amortized cost of ith operation $\stackrel{\text{def}}{=}$ Actual cost of ith operation + $(\phi(i) - \phi(i-1))$

$\Delta\phi_i$ = Potential difference

Amortized cost of n operations $\geq$ Actual cost of n operations

Question 3

- Consider a data structure that is based on a queue with four operations:
  - ENQUEUE(a): Add the element a into the queue
  - DEQUEUE(): Dequeue a single element from the queue
  - DELETE(k): Dequeue k elements from the queue
  - ADD(A): Enqueue all elements in A
- **Claim**: ENQUEUE, DEQUEUE and DELETE run in amortized O(1) time while ADD runs in amortized O(|A|) time.
- Using Potential method, can you show that these time complexities are correct?
- (Please state your potential function.)

Question 4


```
Delete x from T;
n ← n − 1;
If (n = 0)
        free(T);
Else
      If(n = size(T)/2)
    {   T′ ← createTable( n/2);
        copy(T,T′);
        free(T);
        T ← T′
    }
```

Note, T is the dynamic table that supports only deletions.

Using Potential method show that the amortized cost of each Deletion operation is O(1).

(State your potential function.)