Algorithms

Lecture Topic: Amortized Analysis

Anxiao (Andrew) Jiang

Roadmap of this lecture:

1. Amortized analysis by the "Potential Method" technique.

   1.1 Define "Potential Method".

   1.2 Understand "Potential Method" through the example of "Stack Operations".

   1.3 Understand "Potential Method" through the example of "Counter Incrementation".

Technique 3: Potential Method

Starting with an initial data structure D_0, a sequence of n operations occurs.

Technique 3: Potential Method

Starting with an initial data structure D_0, a sequence of n operations occurs.

For each $i = 1,2,\cdots,n,$ let c_i be the actual cost of the i-th operation, and D_i be the data structure that results after applying the i-th operation to data structure D_{i-1}.

Technique 3: Potential Method

Starting with an initial data structure D_0, a sequence of n operations occurs.

For each $i = 1,2,\cdots,n,$ let c_i be the actual cost of the i-th operation, and D_i be the data structure that results after applying the i-th operation to data structure D_{i-1}.

A potential function Φ maps each data structure D_i to a real number $\Phi(D_i)$, which is the potential associated with D_i.

Technique 3: Potential Method

Starting with an initial data structure D_0, a sequence of n operations occurs.

For each $i = 1,2,\cdots,n$, let c_i be the actual cost of the i-th operation, and D_i be the data structure that results after applying the i-th operation to data structure D_{i-1}.

A potential function Φ maps each data structure D_i to a real number $\Phi(D_i)$, which is the potential associated with D_i.

The amortized cost $\hat{c}_i$ of the i-th operation with respect to potential function Φ is defined by $$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

Technique 3: Potential Method

Starting with an initial data structure D_0, a sequence of n operations occurs.

For each $i = 1,2,\cdots,n,$ let c_i be the actual cost of the i-th operation, and D_i be the data structure that results after applying the i-th operation to data structure D_{i-1}.

A potential function Φ maps each data structure D_i to a real number $\Phi(D_i)$, which is the potential associated with D_i.

The amortized cost $\hat{c}_i$ of the i-th operation with respect to potential function Φ is defined by

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

Amortized cost = real cost + change in potential

Technique 3: Potential Method

Starting with an initial data structure D_0, a sequence of n operations occurs.

For each $i = 1, 2, \cdots, n$, let c_i be the actual cost of the i-th operation, and D_i be the data structure that results after applying the i-th operation to data structure D_{i-1}.

A potential function Φ maps each data structure D_i to a real number $\Phi(D_i)$, which is the potential associated with D_i.

The amortized cost $\hat{c}_i$ of the i-th operation with respect to potential function Φ is defined by

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

Amortized cost = real cost + change in potential

Consider the cost of n operations:

$$\sum_{i=1}^{n} \hat{c}_i = \sum_{i=1}^{n} c_i + \Phi(D_i) - \Phi(D_{i-1}) = \sum_{i=1}^{n} c_i + \Phi(D_n) - \Phi(D_0)$$

Technique 3: Potential Method

Starting with an initial data structure D_0, a sequence of n operations occurs.

For each $i = 1, 2, \cdots, n$, let c_i be the actual cost of the i-th operation, and D_i be the data structure that results after applying the i-th operation to data structure D_{i-1}.

A potential function Φ maps each data structure D_i to a real number $\Phi(D_i)$, which is the potential associated with D_i.

The amortized cost $\hat{c}_i$ of the i-th operation with respect to potential function Φ is defined by

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

Amortized cost = real cost + change in potential

Consider the cost of n operations:

$$\boxed{\sum_{i=1}^{n} \hat{c}_i} = \sum_{i=1}^{n} c_i + \Phi(D_i) - \Phi(D_{i-1}) = \boxed{\sum_{i=1}^{n} c_i} + \boxed{\Phi(D_n) - \Phi(D_0)}$$

Technique 3: Potential Method

Amortized cost = real cost + change in potential

$$\boxed{\sum_{i=1}^{n} \hat{c}_i} = \sum_{i=1}^{n} c_i + \Phi(D_i) - \Phi(D_{i-1}) = \boxed{\sum_{i=1}^{n} c_i} + \boxed{\Phi(D_n) - \Phi(D_0)}$$

So if $\Phi(D_n) \geq \Phi(D_0)$, then

$$\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i$$

Quiz question:

1. How is the "Potential Method" different from the "Accounting Method"?

2. What property does the "potential function" need to have?

Roadmap of this lecture:

1. Amortized analysis by the "Potential Method" technique.

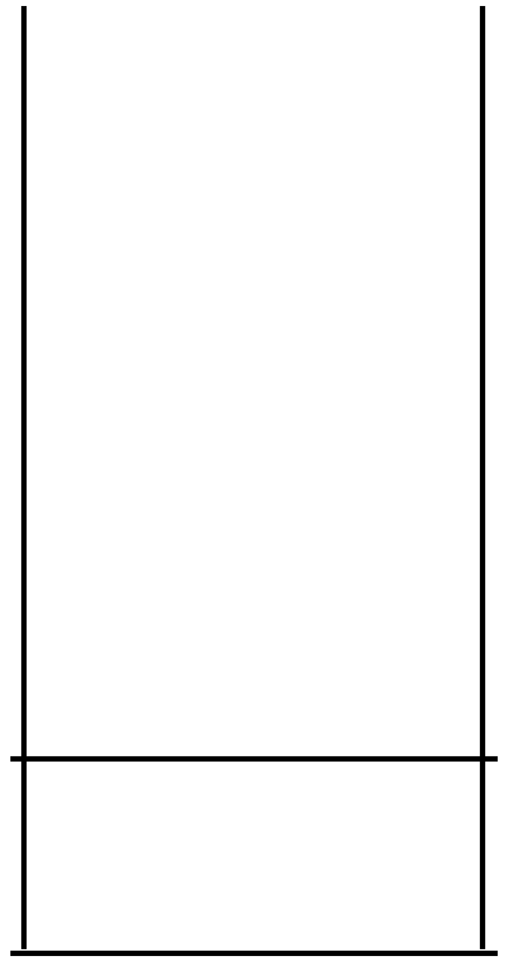
    1.1 Define "Potential Method".

    1.2 Understand "Potential Method" through the example of "Stack Operations".

    1.3 Understand "Potential Method" through the example of "Counter Incrementation".

Technique 3: Potential Method

Example: Stack Operations



Size of Stack: $|S| = 0$

Consider a sequence of n stack operations.
What is a tight upper bound on its total cost ?

Operations:

1) PUSH: push a number into stack

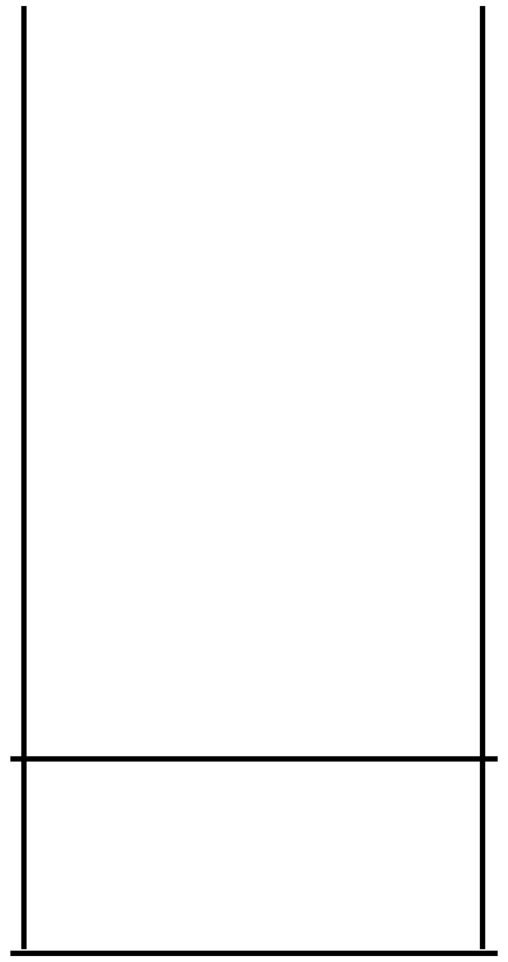
   Real cost: 1

2) POP(k): pop out the top k numbers from stack. If the stack has fewer than k numbers, we pop out all numbers in stack.

   Real cost: $\min\{k, |S|\}$

Technique 3: Potential Method

Example: Stack Operations



Size of Stack: $|S| = 0$

Φ_i : number of objects in the stack after the i-th operation

Operations:

1) PUSH: push a number into stack

   Real cost: 1

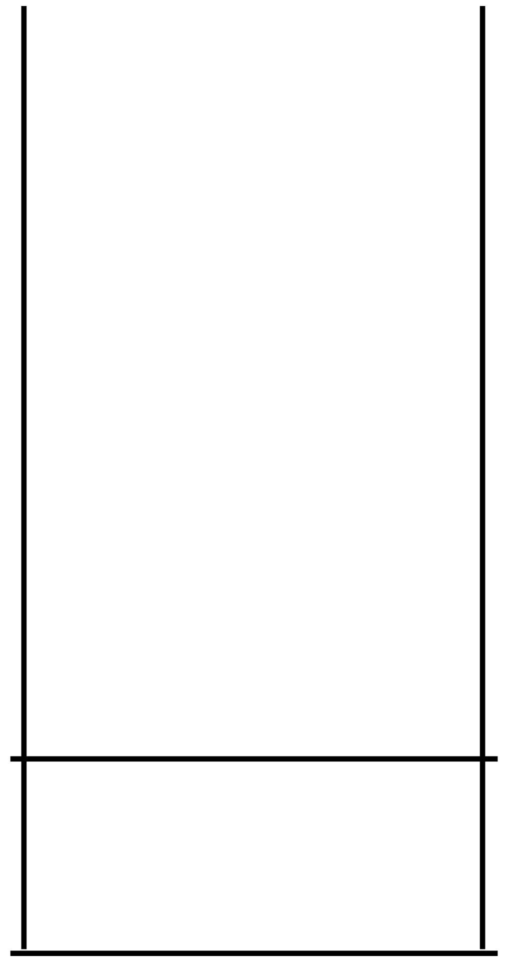
2) POP(k): pop out the top k numbers from stack. If the stack has fewer than k numbers, we pop out all numbers in stack.

   Real cost: $\min\{k, |S|\}$

Consider a sequence of n stack operations.
What is a tight upper bound on its total cost ?

Technique 3: Potential Method

Example: Stack Operations



Size of Stack: $|S| = 0$

Φ_i : number of objects in the stack after the i-th operation

$\Phi_i \geq \Phi_0 = 0$

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i$$

Consider a sequence of n stack operations.
What is a tight upper bound on its total cost ?

Operations:

1) PUSH: push a number into stack

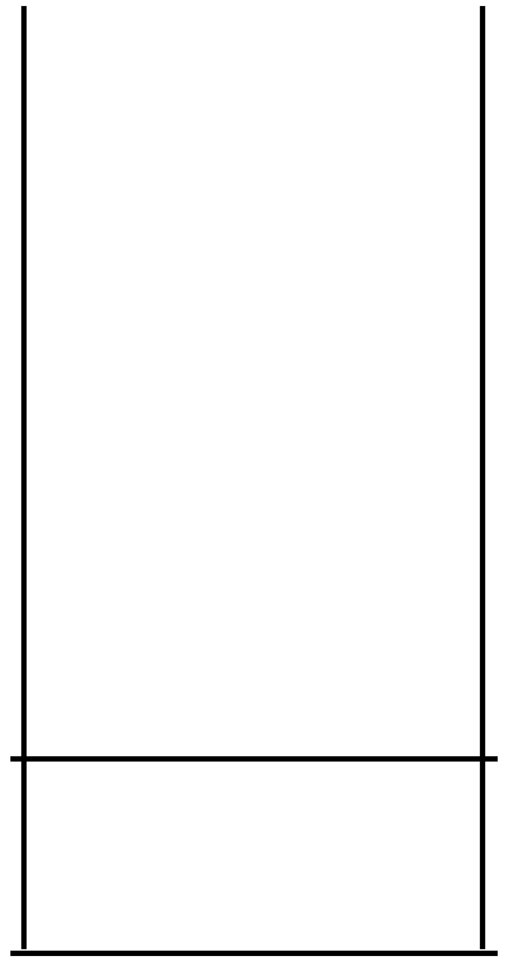
Real cost: 1

2) POP(k): pop out the top k numbers from stack. If the stack has fewer than k numbers, we pop out all numbers in stack.

Real cost: $\min\{k, |S|\}$

Technique 3: Potential Method

Example: Stack Operations



Size of Stack: $|S| = 0$

Φ_i : number of objects in the stack after the i-th operation

$\Phi_i \geq \Phi_0 = 0$

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i$$

Consider a sequence of n stack operations.
What is a tight upper bound on its total cost ?

Operations:

1) PUSH: push a number into stack

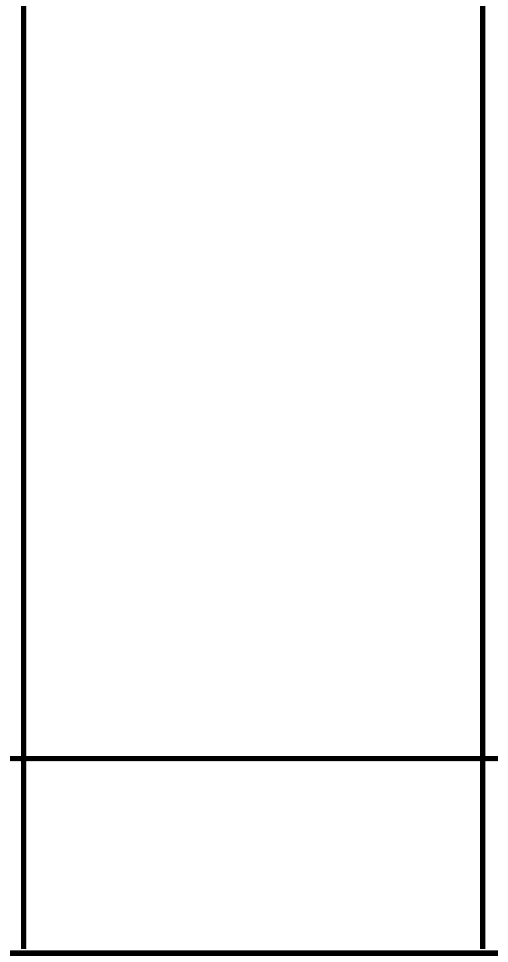
Real cost: 1 Amortized cost: 2

2) POP(k): pop out the top k numbers from stack. If the stack has fewer than k numbers, we pop out all numbers in stack.

Real cost: $\min\{k, |S|\}$

Technique 3: Potential Method

Example: Stack Operations



Size of Stack: $|S| = 0$

Φ_i : number of objects in the stack after the i-th operation

$\Phi_i \geq \Phi_0 = 0$

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i$$

Consider a sequence of n stack operations.
What is a tight upper bound on its total cost ?

Operations:

1) PUSH: push a number into stack

Real cost: 1 Amortized cost: 2

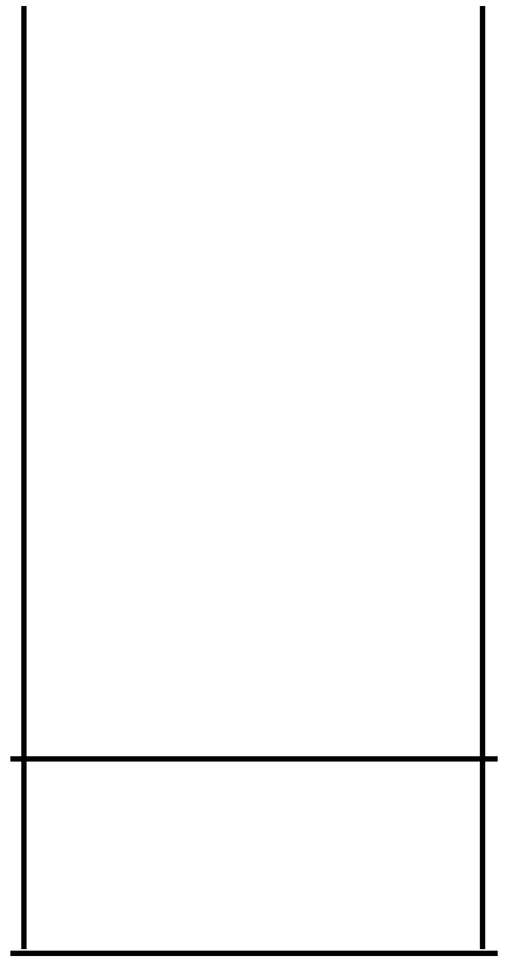
2) POP(k): pop out the top k numbers from stack. If the stack has fewer than k numbers, we pop out all numbers in stack.

Real cost: $\min\{k, |S|\}$

Amortized cost: 0

Technique 3: Potential Method

Example: Stack Operations



Size of Stack: $|S| = 0$

Φ_i : number of objects in the stack after the i-th operation

$\Phi_i \geq \Phi_0 = 0$

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i \leq 2n \qquad O(n)$$

Consider a sequence of n stack operations.
What is a tight upper bound on its total cost ?

Operations:

1) PUSH: push a number into stack

Real cost: 1 Amortized cost: 2

2) POP(k): pop out the top k numbers from stack. If the stack has fewer than k numbers, we pop out all numbers in stack.

Real cost: $\min\{k, |S|\}$

Amortized cost: 0

Quiz question:

1. What was the "potential function" defined in the above example of "stack operations"?

2. Why can the above potential function help us analyze the total cost?

Roadmap of this lecture:

1. Amortized analysis by the "Potential Method" technique.

    1.1 Define "Potential Method".

    1.2 Understand "Potential Method" through the example of "Stack Operations".

    1.3 Understand "Potential Method" through the example of "Counter Incrementation".

Technique 3: Potential Method

Large Binary Counter

0 0 0 0 0 0 0 0 0 0 0

cost: 1

1 0 0 0 0 0 0 0 0 0 1

cost: 2

2 0 0 0 0 0 0 0 0 1 0

cost: 1

3 0 0 0 0 0 0 0 0 1 1

cost: 3

4 0 0 0 0 0 0 0 1 0 0

cost: 1

5 0 0 0 0 0 0 0 1 0 1

cost: 2

6 0 0 0 0 0 0 0 1 1 0

cost: 1

7 0 0 0 0 0 0 0 1 1 1

cost: 4

8 0 0 0 0 0 0 1 0 0 0

cost: 1

9 0 0 0 0 0 0 1 0 0 1

cost: 2

10 0 0 0 0 0 0 1 0 1 0

cost: 1

11 0 0 0 0 0 0 1 0 1 1

Cost of incrementing counter:

Number of bits that are changed.

What is the total cost of n increments?

Technique 3: Potential Method

Φ_i : number of 1s in the counter after the i-th operation

Large Binary Counter

0	0 0 0 0 0 0 0 0 0 0	
		cost: 1
1	0 0 0 0 0 0 0 0 0 1	
		cost: 2
2	0 0 0 0 0 0 0 0 1 0	
		cost: 1
3	0 0 0 0 0 0 0 0 1 1	
		cost: 3
4	0 0 0 0 0 0 0 1 0 0	
		cost: 1
5	0 0 0 0 0 0 0 1 0 1	
		cost: 2
6	0 0 0 0 0 0 0 1 1 0	
		cost: 1
7	0 0 0 0 0 0 0 1 1 1	
		cost: 4
8	0 0 0 0 0 0 1 0 0 0	
		cost: 1
9	0 0 0 0 0 0 1 0 0 1	
		cost: 2
10	0 0 0 0 0 0 1 0 1 0	
		cost: 1
11	0 0 0 0 0 0 1 0 1 1	

Cost of incrementing counter:
Number of bits that are changed.

What is the total cost of n increments?

Technique 3: Potential Method

Φ_i : number of 1s in the counter after the i-th operation

$$\Phi_i \geq \Phi_0 = 0$$

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i$$

Large Binary Counter

0	0 0 0 0 0 0 0 0 0 0	
		cost: 1
1	0 0 0 0 0 0 0 0 0 1	
		cost: 2
2	0 0 0 0 0 0 0 0 1 0	
		cost: 1
3	0 0 0 0 0 0 0 0 1 1	
		cost: 3
4	0 0 0 0 0 0 0 1 0 0	
		cost: 1
5	0 0 0 0 0 0 0 1 0 1	
		cost: 2
6	0 0 0 0 0 0 0 1 1 0	
		cost: 1
7	0 0 0 0 0 0 0 1 1 1	
		cost: 4
8	0 0 0 0 0 0 1 0 0 0	
		cost: 1
9	0 0 0 0 0 0 1 0 0 1	
		cost: 2
10	0 0 0 0 0 0 1 0 1 0	
		cost: 1
11	0 0 0 0 0 0 1 0 1 1	

Cost of incrementing counter:
Number of bits that are changed.

What is the total cost of n increments?

Technique 3: Potential Method

Φ_i : number of 1s in the counter after the i-th operation

$$\Phi_i \geq \Phi_0 = 0$$

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i$$

Consider the i-th operation:

? ? ?⋯0 1 1 ⋯1

⬇

? ? ?⋯1 0 0 ⋯0

Large Binary Counter

0	0 0 0 0 0 0 0 0 0 0	cost: 1
1	0 0 0 0 0 0 0 0 0 1	cost: 2
2	0 0 0 0 0 0 0 0 1 0	cost: 1
3	0 0 0 0 0 0 0 0 1 1	cost: 3
4	0 0 0 0 0 0 0 1 0 0	cost: 1
5	0 0 0 0 0 0 0 1 0 1	cost: 2
6	0 0 0 0 0 0 0 1 1 0	cost: 1
7	0 0 0 0 0 0 0 1 1 1	cost: 4
8	0 0 0 0 0 0 1 0 0 0	cost: 1
9	0 0 0 0 0 0 1 0 0 1	cost: 2
10	0 0 0 0 0 0 1 0 1 0	cost: 1
11	0 0 0 0 0 0 1 0 1 1	

Cost of incrementing counter:
Number of bits that are changed.

What is the total cost of n increments?

Technique 3: Potential Method

Φ_i : number of 1s in the counter
after the i-th operation

$$\Phi_i \geq \Phi_0 = 0$$

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i$$

Consider the i-th operation:

? ? ?⋯0 1 1 ⋯1

$\Downarrow$

? ? ?⋯1 0 0 ⋯0

Amortized cost: 2

Large Binary Counter

0 0 0 0 0 0 0 0 0 0 0
cost: 1
1 0 0 0 0 0 0 0 0 0 1
cost: 2
2 0 0 0 0 0 0 0 0 1 0
cost: 1
3 0 0 0 0 0 0 0 0 1 1
cost: 3
4 0 0 0 0 0 0 0 1 0 0
cost: 1
5 0 0 0 0 0 0 0 1 0 1
cost: 2
6 0 0 0 0 0 0 0 1 1 0
cost: 1
7 0 0 0 0 0 0 0 1 1 1
cost: 4
8 0 0 0 0 0 0 1 0 0 0
cost: 1
9 0 0 0 0 0 0 1 0 0 1
cost: 2
10 0 0 0 0 0 0 1 0 1 0
cost: 1
11 0 0 0 0 0 0 1 0 1 1

Cost of incrementing counter:
Number of bits that are changed.

What is the total cost of n increments?

Technique 3: Potential Method

Φ_i : number of 1s in the counter after the i-th operation

$\Phi_i \geq \Phi_0 = 0$

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i \leq 2n \qquad O(n)$$

Consider the i-th operation:

? ? ?⋯0 1 1 ⋯1

↓

? ? ?⋯1 0 0 ⋯0

Amortized cost: 2

Large Binary Counter

0   0 0 0 0 0 0 0 0 0 0
cost: 1
1   0 0 0 0 0 0 0 0 0 1
cost: 2
2   0 0 0 0 0 0 0 0 1 0
cost: 1
3   0 0 0 0 0 0 0 0 1 1
cost: 3
4   0 0 0 0 0 0 0 1 0 0
cost: 1
5   0 0 0 0 0 0 0 1 0 1
cost: 2
6   0 0 0 0 0 0 0 1 1 0
cost: 1
7   0 0 0 0 0 0 0 1 1 1
cost: 4
8   0 0 0 0 0 0 1 0 0 0
cost: 1
9   0 0 0 0 0 0 1 0 0 1
cost: 2
10   0 0 0 0 0 0 1 0 1 0
cost: 1
11   0 0 0 0 0 0 1 0 1 1

Cost of incrementing counter:
Number of bits that are changed.

What is the total cost of n increments?

Quiz question:

1. What was the “potential function” defined in the above example of “counter incrementation”?

2. Why can the above potential function help us analyze the total cost?