Amortized Analysis Algorithms

D.S. \[ o_1, o_2, o_3, \ldots, o_n \] \[ \sum_{i=1}^{n} c(o_i) \]

Worst case \[ c(o_i) \]

Sum of all actual costs of ops \[ = \text{AMORTIZED COST} \]

Worst case avg cost per operation

Stack

$2$ $0$

Accounting Method

Push \[ O(1) \]

Pop \[ O(1) \]

\[ O(n) \]

Multiplying \( O(n) \) = removes top hex from the stack

Worst case \[ \sum_{i=1}^{n} c(o_i) \] at most \( n \) pushes \( \Rightarrow O(n) \)

Worst case cut \( O(n) \)

Worst case cut

Incrementing a Counter
17.1 Aggregate analysis

<table>
<thead>
<tr>
<th>Counter value</th>
<th>Total cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>18</td>
</tr>
<tr>
<td>11</td>
<td>19</td>
</tr>
<tr>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>13</td>
<td>23</td>
</tr>
<tr>
<td>14</td>
<td>25</td>
</tr>
<tr>
<td>15</td>
<td>26</td>
</tr>
<tr>
<td>16</td>
<td>31</td>
</tr>
</tbody>
</table>

\[ \left( n + \frac{n}{2} + \frac{n}{4} + \ldots \right) \leq n \left( 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots \right) \]

\[ \leq 2n \]

\[ \text{\$2/\text{increment}} \]

**Figure 17.2** An 8-bit binary counter as its value goes from 0 to 16 by a sequence of 16 INCREMENT operations. Bits that flip to achieve the next value are shaded. The running cost for flipping bits is shown at the right. Notice that the total cost is always less than twice the total number of INCREMENT operations.

operations on an initially zero counter causes \( A[1] \) to flip \( \lfloor n/2 \rfloor \) times. Similarly, bit \( A[2] \) flips only every fourth time, or \( \lfloor n/4 \rfloor \) times in a sequence of \( n \) INCREMENT operations. In general, for \( i = 0, 1, \ldots, k - 1 \), bit \( A[i] \) flips \( \lfloor n/2^i \rfloor \) times in a sequence of \( n \) INCREMENT operations on an initially zero counter. For \( i \geq k \), bit \( A[i] \) does not exist, and so it cannot flip. The total number of flips in the sequence is thus

\[
\sum_{i=0}^{k-1} \left\lfloor \frac{n}{2^i} \right\rfloor < n \sum_{i=0}^{\infty} \frac{1}{2^i} = 2n
\]

by equation (A.6). The worst-case time for a sequence of \( n \) INCREMENT operations on an initially zero counter is therefore \( O(n) \). The average cost of each operation, and therefore the amortized cost per operation, is \( O(n)/n = O(1) \).
"Ratchet account" - POTENTIAL FUNCTION

\[ \Phi(t) = \text{money in bank} \]

\[ \Phi(\text{stack}) = \# \text{ of items on the stack} \]

\[
\text{Actual cost} = \text{Amortized cost} + \Phi(\text{before}) - \Phi(\text{after})
\]

\[
\text{Amortized cost} = \text{Actual cost} + \Phi(\text{after}) - \Phi(\text{before})
\]

\[
\Delta \Phi = \Phi(\text{after}) - \Phi(\text{before}) = \Delta
\]

\[
\text{Cost (sequence)} = \sum \text{actual cost}
\]

\[
= \sum \left( \text{Amortized cost} + \Phi(\text{before}) - \Phi(\text{after}) \right)
\]

\[
= \left( \sum \text{Amortized cost} \right) + \sum \left[ \Phi(\text{before}) - \Phi(\text{after}) \right]
\]

Initial - Initial = Final - Final
\[ \text{Total actual cost} = \text{Total amortized cost} + \Delta \overline{\Phi} \]

\[ \overline{\Phi}(\text{stack}) = \# \text{ of items on the stack} \]

Annual cost = $2/\text{perk} + $6/\text{per unit} multiply

2n

Amortized cost \leq 2n

\[ \Delta \overline{\Phi} \leq 0 \]

\[ \overline{\Phi}(\text{current}) = \# \text{ of 1-bit} \]

Amortized cost of initial = $2

\[ \sum \text{actual cost} \leq \sum \text{amortized cost} + \Delta \overline{\Phi} \]

\[ \sum \text{actual cost} \leq 2n \]

\[ \text{actual cost} = 1 + \# \text{ of 1-bit carry appropriate} \]

\[ \frac{2n}{0} \leq 0 \]
then compute amortized costs of operation
\[
\text{cost} + \Delta \Phi
\]

\begin{itemize}
\item \text{tables}
\end{itemize}

\begin{itemize}
\item insertion \quad \text{insertion} \quad O(1)
\item doubling insertion \quad \Theta(\text{table size})
\end{itemize}