§3 Worst-Case vs. Amortized

Motivating example: Repeated binary increment,
#bit flips when counting to n?   Inc **& decrement** ?

```
      0
      1
     10
     11
    100
    101
    ……
111…111
```

<u>One</u> increment of value $<n$: $O(\log n)$

n increments: $\sum_{j<n} \log j = \Theta(n \cdot \log n)$

Bit #0 flips $n/2$ times,

bit #1 incurs $n/4$ flips, bit #2: $n/4$ flips,

$\sum \leq 2n = \Theta(n)$: constant (2) per **`inc`**

Potential method of analysis:

Let c_j denote cost of j-th operation,

Φ_j:=#**1**s in counter after j-th op. ⇔ before $(j+1)$-st op.

$\sum_{1\leq j\leq n} c_j/n \;\leq\; \max_j\,(c_j+\Phi_j-\Phi_{j-1}) + (\Phi_0-\Phi_n)/n \quad = 2 - \Phi_n/n$

§3 Amortized Analysis

Motivating example: Repeated binary increment,
#bit flips when counting to n?   Inc **& decrement** ?

Definition: Fix an implementation $\mathcal{A}$ of abstract data type $\mathcal{D}$ with methods $M_1, M_2,\ldots,M_k$.

Let $T(n)$ denote the <u>worst-case</u> cost of <u>any</u> n-element sequence <u>C</u> of calls $C_1,C_2,\ldots,C_n \in \{M_1,\ldots,M_k\}$.

Then the **amortized cost** of $\mathcal{A}$ is defined as $T(n)/n$.

Don't confuse:
- amortized cost
- average-case cost
- expected cost

Potential method of analysis:

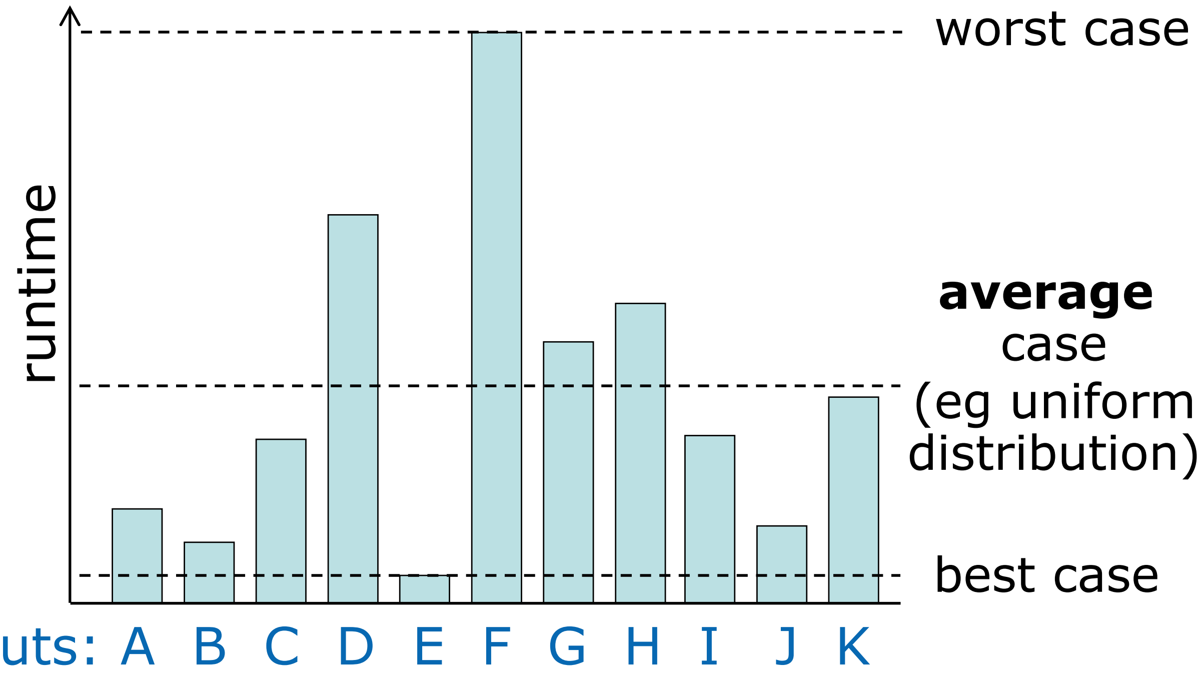
Let c_j denote cost of j-th operation,

Conceive Φ_j such that right hand side is „small"

$\sum_{1\leq j\leq n} c_j/n \;\leq\; \max_j\,(c_j+\Phi_j-\Phi_{j-1}) + (\Phi_0-\Phi_n)/n$

Worst, Average-Case, Expected, Amortized

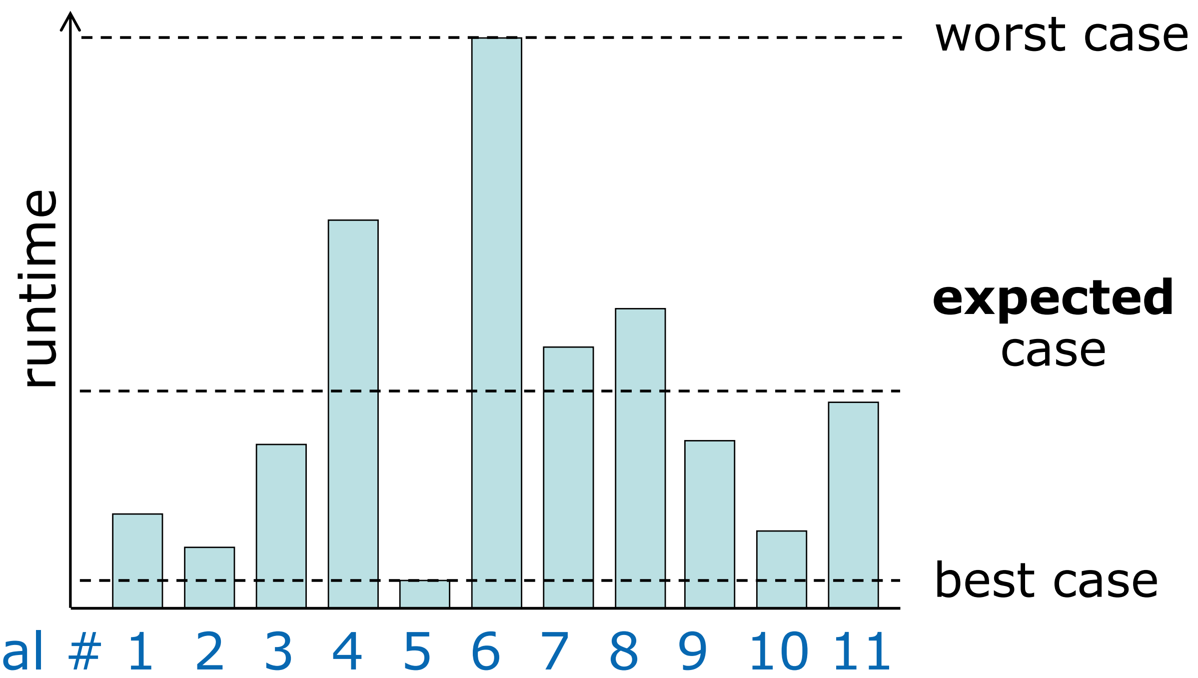
Algorithm $\mathcal{A}$, fixed input size n:



asymptotically as $n \to \infty$

Worst, Average-Case, Expected, Amortized

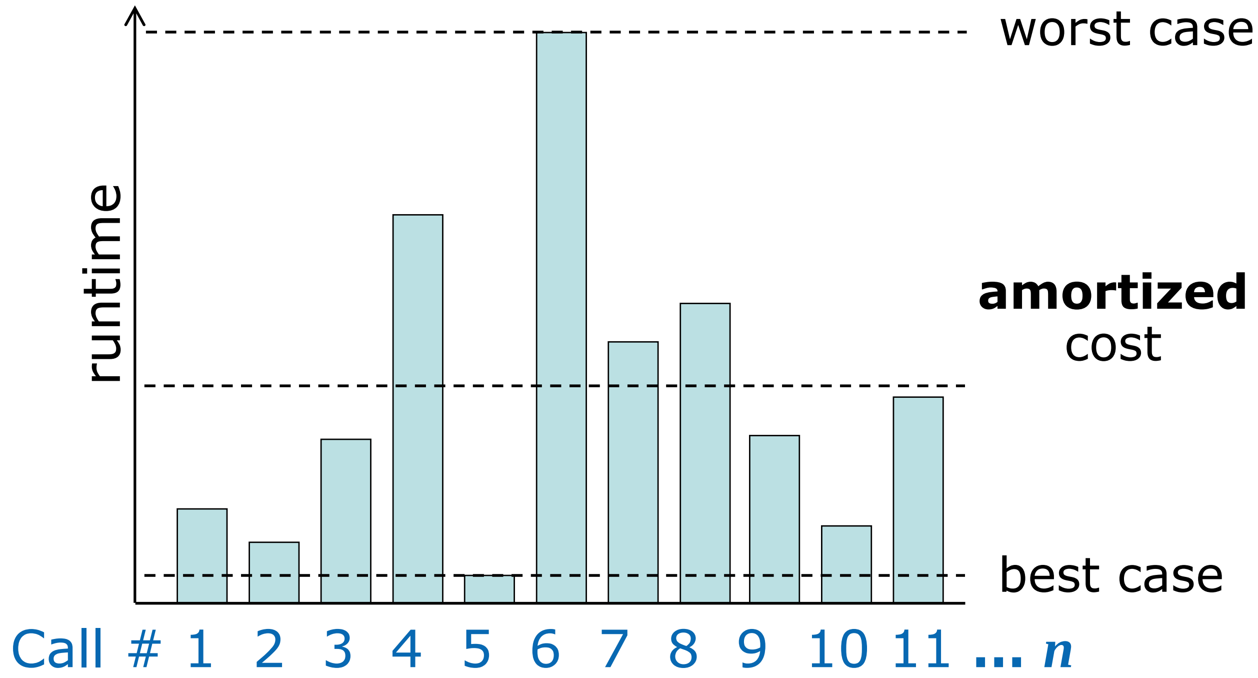
Randomized algorithm $\mathcal{A}$, fixed input X:



worst-case over all X of length n asympt. as $n \to \infty$

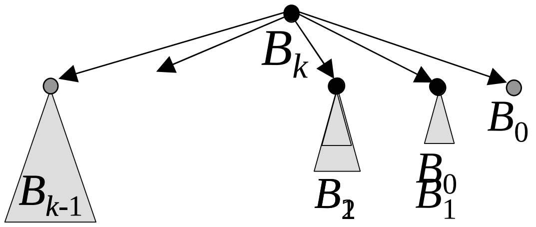
Worst, Average-Case, Expected, Amortized

Data type $\mathcal{D}$, method M



Methods $M_1,\ldots M_k$: worst-case over seq. of length n

§3 *Relaxed* Binomial Trees



"Mark" • indicates child #j may have order $=j$-2

A *relaxed* Binomial Tree of order $k\geq 1$ consists of a root with k children, j^{th} (j=1…k) being a relaxed binom.tree of order $\geq j$-2

Lemma: A relaxed Binomial Tree of order k has $\geq F_{k+2} \geq \Omega(1.6^k)$ nodes

Merge

Prune

Proof by induction + homework:

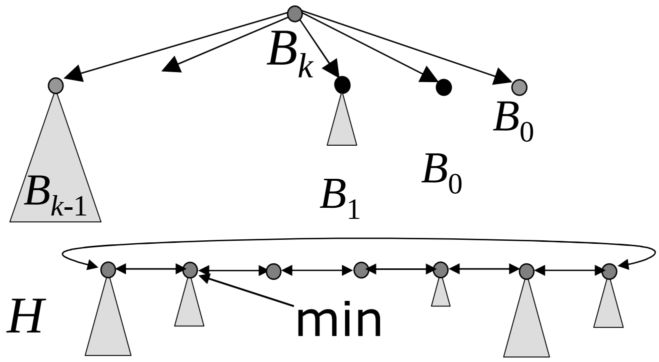
$$1 + \underbrace{F_1}_{1} + \underbrace{F_2}_{1} + \ldots + F_k = F_{k+2} \geq \varphi^k$$

$\varphi := (1+\sqrt{5})/2 > 1.6$

Fibonacci no.s F_k

§3 Fibonacci Heaps

KAIST
CS500 M. Ziegler



A *relaxed* Binomial Tree of order $k \geq 1$ consists of a root with k children, j^{th} (j=1…k) being a **relaxed** binom.tree of order ≥ j-2

- child j may have order j-2

Lemma: A relaxed Binomial Tree of n nodes has order $k \leq O(\log n)$

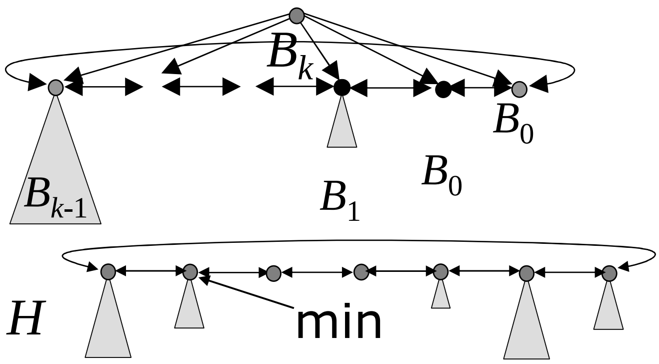
A Fibonacci Heap H is a list of heap-ordered relaxed binomial trees with pointer to the min.

Extract min.key: $O(\log n)$
Decrease key: $O(1)$
Merge two Fib.heaps: $O(1)$
Insert element: $O(1)$
Create 1-elem.Fib.heap: $O(1)$

amortized cost

§3 Extract Minimum

KAIST
CS500 M. Ziegler



A *relaxed* Binomial Tree of order $k \geq 1$ consists of a root with k children, j^{th} (j=1…k) being a **relaxed** binom.tree of order ≥ j-2

- child j may have order j-2

Lemma: A relaxed Binomial Tree of n nodes has order $k \leq O(\log n)$

A Fibonacci Heap H is a list of t heap-ordered relaxed binomial trees with pointer to the min.

new?

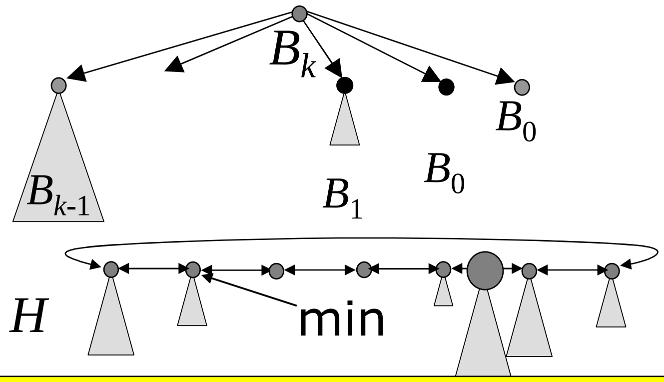
Extract min.key: $O(\log n)$ amortized cost

- Delete target of min.pointr
- Merge two Fibonacci heaps.
- **Consolidate** s.t. each tree order k occurs only once!

bucket sort

$c_j + \Phi_j - \Phi_{j-1} \overset{!}{\leq} O(\log n)$ | Potential $\Phi \overset{!}{:=} \Theta(t)$ | $\Phi_0 = 0, \Phi_n \geq 0$

§3 Decrease Key



A *relaxed* Binomial Tree of order $k \geq 1$ consists of a root with k children, j^{th} (j=1…k) being a relaxed binom.tree of order $\geq j$-2

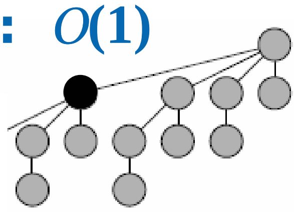
- child j may have order j-2

Lemma: A relaxed Binomial Tree of n nodes has order $k \leq O(\log n)$

A Fibonacci Heap H is a list of t heap-ordered relaxed binomial trees with pointer to the min.

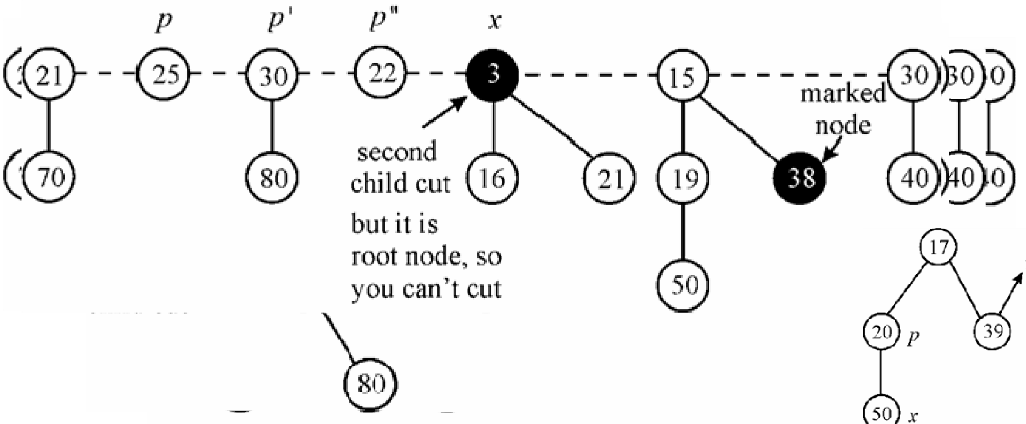
Decrease key: $O(1)$ amortized cost

- cut subtree
- **mark** parent
- if already **marked**: reset, cut & cascade up

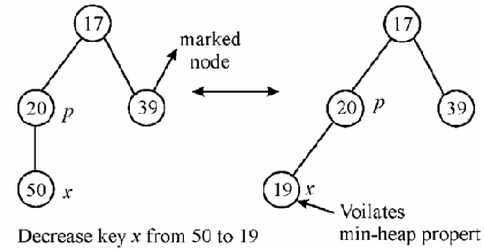


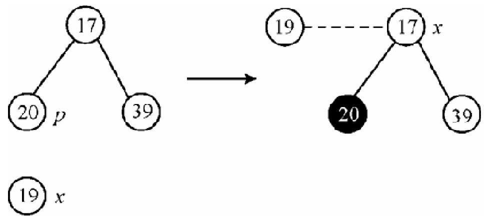
$c_j + \Phi_j - \Phi_{j\text{-}1} \leq O(1)$ Potential $\Phi := \Theta(t+2m)$ $\Phi_0 = 0,\ \Phi_n \geq 0$

§3 Cuts, Marks, and Cascading



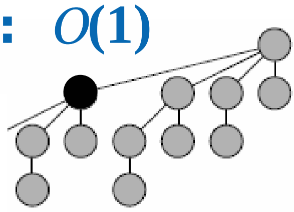
(decrease key of x from 42 to 21)





Decrease key: $O(1)$

- cut subtree
- **mark** parent
- if already **marked**: reset, cut & cascade up



$c_j + \Phi_j - \Phi_{j\text{-}1} \leq O(1)$ Potential $\Phi := C\cdot(t+2m)$ $\Phi_0 = 0,\ \Phi_n \geq 0$