CSE 332: Data Structures & Parallelism

Lecture 6: Amortization

Ruth Anderson

Autumn 2025

Administrative

- EX0 – Closes tonight
- EX01 – Due TONIGHT, Monday 10/06
- EX02 – On Priority Queues, Due Friday Oct 10
- EX03 – On Recurrences, coming soon!
- “Meet the Staff” activity
  - Sometime during the first 4 weeks of class, visit a CSE 332 office hour (in person or on zoom)
- Lecture MegaThread in Ed Discussion

Today

- Finish Binary Min Heap implementation (Slides from Friday)
  - Buildheap
- Amortization
- ~~Recurrences~~

Amortization

- How much does housing cost per day in Seattle?
- Well, it depends on the day.

- The day rent is due, it’s $1800.
- The other days of the month it’s free.

Amortization

- Amortization is an **accounting analysis**. It's a way to reflect the fact that even though the "first of the month" is very expensive, the reason that it's very expensive is that it's taking on responsibility for all the other days.
- If we distributed the cost equally across the days, (because all days *should* be equally responsible), we "amortize" the cost.

Amortization

AMORTIZED

- It costs $1800/month (which we pay once)
- So the cost per day is $\frac{1800}{30} = 60.$
- Good answer if the question is “what does my daily pay need to be to afford housing?“

UNAMORTIZED

- On the first it costs $1800.
- Every other day of the month it costs $0
- Good answer if the question is “how much do I need to keep in my bank account so it doesn’t get overdrawn?”

Amortization

- What's the worst case for `enqueue` into an array-based queue?
  - The running time is $O(n)$ when we need to resize, and $O(1)$ otherwise.
- Is $O(n)$ a good description of the worst-case behavior?

- Imagine you said:
  "In the worst-case, rent costs $1800 per day. There are 30 days this month, so I need to set aside $30 \cdot 1800 = \$54{,}000$ in my budget; that's the worst-case for the month"
- Or you said on Apr. 30, "rent costs $60/day, it's fine that I have only $70 in my bank account"
  - Both of these are silly!

Amortization

AMORTIZED

- It takes $O(n)$ time to resize once, the next $n-1$ calls take $O(1)$ time each.
- So the cost per operation is $\frac{O(n)+[n-1]O(1)}{n} = O(1)$
- Good answer if the question is "what will happen when I do many insertions in a row?"

UNAMORTIZED

- The resize will take $O(n)$ time. That's the worst thing that could happen.
- Good answer if the question is "how long might one (unlucky) user need to wait on a single insertion?"

Amortization

- Why do we double? Why not increase the size by 1 each time we fill up?

Amortization vs. Average-Case

- Amortization and “average/best/worst” case are independent properties (you can have un-amortized average-case, or amortized worst-case, or un-amortized worst-case, or …).
- Average case asks: “if I selected a possible input on random, how long would my code take?” (compare to worst-case: “if I select the worst value…”)

- Amortized or not is “do we care about how much our bank account changes on one day or over the entire month?” (do we care about the running time of individual calls or only what happens over a sequence of them?)

Why use (or don't use) amortized analysis?

Worst Case	Amortized	Unamortized
`Enqueue` (circular array)	$\Theta(1)$	$\Theta(n)$

- The appropriate analysis depends on your situation (and often it's worth knowing both).
- A common use of data structures is as part of an algorithm.
  - E.g., I'm trying to process everything in a data set, I insert everything into the data structure, remove them one at a time.
  - In that case, we almost always want amortized analysis (we care about when the full analysis is done, not when we go from 49% done to 50% done).
- But sometimes you care about individual calls
  - Your data structure is feeding another process that the user is watching in real-time.

Reminder: O, Ω, Θ vs. Best, Worst, Average

- It's a common misconception that $\Omega()$ is "best-case" and $O()$ is "worst-case". This is a misconception!!
- $O()$ says "the complexity of this algorithm is at most" (think $\leq$)
- $\Omega()$ says "the complexity of this algorithm is at least" (think $\geq$)
- You can use $\leq$ on worst-case or best case; you can use $\geq$ on worst-case or best-case.
- Best/Worst/Average say "what function f am I analyzing?"
- O, Ω, Θ say "let me summarize what I know about f, it's $\leq, \geq, =$…"