Today: More integers
- predecessor lower bound review
- signature/packed sorting review
  ~ mergesort
- packed quicksort

Predecessor LB: \( \min \{ \log_a w, \log_b n^3 \} \)
  
  Alice's msg. = \( \log(\text{space}) \)
  \( w = \text{Bob's msg.} \)

- warning: lower bounds are hard!
- idea: construct tough instance (distribution)

1. split words into \( k \) chunks, make data identical in chunks \(< i \)
   \( \Rightarrow \) predecessor determined by chunk \( i \)
   but query (Alice) doesn't know \( i \)
   & make chunk \( i \) bad via \( \odot \), \( w' = w/k \)

2. split \( n \) items into \( k' \) chunks, make \( i \)th chunk items start \( \text{bin}(i) \)
   \( \Rightarrow \) which subproblem determined by \( i = \text{lead} \)
   \( lg k \) bits of query (not known to DS/Bob)
   & make chunk \( i \) bad via \( \odot \), \( n' = n/k' \)
   \( w' = w - lg k' \)
Round elimination argument:
- consider t-round communication for $\mathbb{1}$
- Alice's first message has $\approx \frac{a}{k}$ useful bits
  - Bob can guess them with prob. $\frac{1}{2^{a/k}}$
  - error prob. $1 - \frac{1}{2^{a/k}} \approx \frac{a}{k}$... actually $\sqrt{a/k}$
- left with Bob-first comm. for $\mathbb{2}$
- eliminate Bob's message, error prob. $+\sqrt{b/k}$
- left with Alice-first comm. for $\mathbb{1}$
- after $t$ such round eliminations,
  - left with $0$-message protocol for $\mathbb{1}$

$\Rightarrow$ error prob. must be $\geq \frac{1}{2}$
  if $n' \geq a$ & $w' \geq 1$ (non-trivial instance)
  i.e. $t \leq \min \{ \log_k w, \log_k n^2 \}$
- set $k = a t^2$ & $k' = b t^2$

$\Rightarrow$ if $t \leq \min \{ \log_{a t^2} w, \log_{b t^2} n^2 \}$
  $t = O(\lg n)$ & $a \geq \lg n^2 = O(a^3)$
  $b t^2 = O(b^3)$ \(\Rightarrow\) $t = O(\lg w) = O(\lg b)$

then error $= t \left( \frac{\sqrt{a/k'}}{t} + \frac{\sqrt{b/k'}}{t} \right)$
  $= t \left( \frac{1}{t} + \frac{1}{t} \right)$
  $= \frac{1}{3}$ with appropriate constants
  $\geq \frac{1}{2} \text{ CONTRADICT} \blacksquare$
Packed sorting: $n$ $b$-bit ints. with $w = \Omega(b \log n \log \log n)$

= mergesort with ints. packed in $n/k$ words

$\Rightarrow T(n) = 2T(n/2) + O(n \cdot \log k)$

= $O(n \cdot \log k \cdot \log n)$

= $O(n)$

- merge via bitonic sorting + bit tricks

Signature sort: $O(n)$ time for $w \geq \log^{2+\varepsilon} n \cdot \log \log n$

- break integers into $\log^{\varepsilon} n$ chunks of $\log^2 n \cdot \log \log n$

- hash each chunk to $\log n$ bits

$\Rightarrow \log^{1+\varepsilon} n$-bit signature for each integer

- sort them via packed sorting

- fix chunk order of each node in trie by recursively sorting (node, chunk, edge index)

$\Rightarrow$ get correct permutation on edges

$\Rightarrow b' = b / \log^{\varepsilon} n + O(\log n)$

$\Rightarrow$ after $1/\varepsilon$ recursions, $b' = O(w / \log^{1+\varepsilon} n)$

= $O(w / \log n \log \log n)$

$\Rightarrow$ can use packed sorting to finish
Problem: Packed quicksort \( w = \Omega(b \log h \log n) \)

- quick sort with ints, packed in \( n/k \) words
- choose partition element \( x \) (e.g., random)
- partition array into \( \leq x \) & \( > x \)
- recursively sort
- concatenate

1. partition 1 word with \( k \) elements:

\[
\emptyset \ a_1 \ \emptyset \ a_2 \ \emptyset \ \emptyset \ \emptyset \ \emptyset \ a_k
\]

into 2 words storing elts. \( \leq x \) & \( > x \)

in \( O(1) \) time on word RAM

2. given 2 words with \( j_1, j_2 \leq k/4 \) elts. spread out:

\[
\emptyset \ \emptyset \ \widehat{1} \ a_1 \ \; \emptyset \ \emptyset \ a_2 \ \emptyset \ \emptyset \ \widehat{1} \ a_3 \ \emptyset \ \emptyset
\]

combine into 1 word with all \( j_1 + j_2 \) elts.

in \( O(\log k) \) expected time on word RAM

2'. given 1 word with \( j \leq k \) elements spread out:

\[
\emptyset \ \emptyset \ \widehat{1} \ a_1 \ ; \emptyset \ a_2 \ \emptyset \ \emptyset \ \widehat{1} \ a_3 \ \emptyset \ \emptyset
\]

compactify to right so that

\[ \geq \text{constant fraction density} \]

in \( \rho \approx \log k \) time?

3. partition \( O(\frac{n}{k}) \) words each with \( \Theta(k) \) elts.

\[ \leq x \]

& \( O(\frac{n}{k}) \) words each with \( \Theta(k) \) elts. \( > x \)

in \( O(\frac{n}{k} \log k) \) expected time on word RAM