Aston Notes: Wildcard String Matching
- wildcard = single arbitrary character
- all-to-all matching: many patterns & texts
  - various special cases of interest
- offline: [Cole et al.]
  - $O(m + 2^k \log \log n + \text{output})$ query
  - $O(n \log n + x(k + \log x)^k/k!)$ preprocessing.
- online: [Fischer & Paterson 1974]
  - convolution of characters' binary codes
  - $O(n \log m \log |\Sigma|)$ query
  - wildcards in text & pattern
- online improvements:
  - $O(n \log n)$ Monte Carlo [Indyk 1998]
  - $O(n \log m)$ Monte Carlo [Kalai 2002]
  - $O(n \log m)$ deterministic [Cole & Harihan 2002]
Meshkat Farvokhzadi: Entropy & Information Rate in email

- Kolmogorov complexity of string \( x = \) length of shortest program outputting \( x \)
  - uncomputable
  - compression = upper bound

- 650,000 email corpus
  but anonymized & hashed as word set for privacy
  \( \Rightarrow \) can't do Lempel-Ziv, can do Huffman
  - but don't really want different codebook per message ~ use global

- use Enron email database to measure these effects of hashing

- measure change in person's entropy, info. rate over time

[RESEARCH/IMPLEMENTATION]
Kah Keng Tay: Sublogarithmic Nearest Neighbor
- Voronoi diagram as black box
- split into vertical slabs thru Voronoi vxs.
- binary search for slab, within slab
- new approach: $y$-like $y$-fast
  - quad trees + hashing + Voronoi + indirection
  - $O(n \log^2 n)$ preproc., $O(n)$ space
  - $O(\log \log n)$ expected query
    assuming uniform point set
Aaron Bernstein: Distance Sensitivity Oracles

- query: distance from u to v avoiding a vertex x
- motivation: Vickery pricing
- $\tilde{O}(mn^2)$ time, $\tilde{O}(n^5)$ space
- $O(mn^{1.5})$ time, $\tilde{O}(n^{3.5})$ space
- **new**: $\tilde{O}(mn^{1.5})$ time, $\tilde{O}(n^3)$ space
  - based on random sampling
  - also: $\tilde{O}(n^3)$ time (good for dense graphs)
    $O(n^3 \sqrt{m})$ time for unweighted graph

**THEORY**
Alex Schwendner: Highly Connected Components

- $k$-vertex-connectivity query: are there $k$ vertex-disjoint paths from $u$ to $v$
- $O(lg^4 n)$ updates for 2-edge [Holm et al.]
  - maintain nontree edges in levels too
  - each induces a cycle
- $O(lg^5 n)$ updates for 2-vertex [ibid]
- 3 & 4: much slower
- future: poly$lg$ for $k = O(1)$?

[Survey]
Matthew Hofmann & Aditya Rotham: implementing predecessor

- balanced BSTs
- van Emde Boas
- x-fast & y-fast trees
- implemented FKS hashing + optimizations
  - VEB tweak: scan summary linearly if small
    - round up to next $2^k$
- test on 32-bit architecture, pretending to have $w = 8, 16, 32$
  - VEB is factor 40-60x slower than BST!
    & slows when $n \uparrow$ because of min opt.
- x-fast & y-fast in progress

IMPLEMENTATION
Boris Alexeev: Perfect Heaps
- motivation: MSTs
- soft heaps: [Chazelle]
  - after n operations, can corrupt Θn elts.
  - O(1) time meld, delete, findmin,
    O(lg 1/ε) insert
  - fastest deterministic MST alg. O(nα(n))
  - # inversions < εn² => sort in linear time
  - partitioning into intervals of ≤ 3εn², linear time
- new: “perfect heaps”
  - simple: trees always perfect
  - almost as good as Fibonacci
  - decrease-key... O(1) in avg. case

THEORY
Marti Bolivar: Scene Recognition
- Log-polar mapping ~ focus on center of img.
- LSH to find features, similarity, newness

IMPLEMENTATION