Research Proposal: A data structure to support the simulation of random events

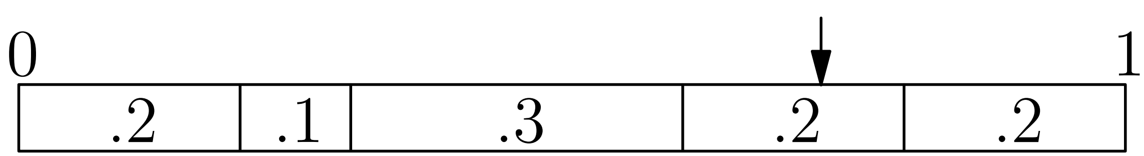
Hubert Chan

Outline

- the random object selection problem
- reaction-diffusion equation (motivation)
- problem description
- related problems:
  - selectable partial sums
  - optimal search trees/coding systems
- research goals and approaches

Random object selection problem

- Given:
  - n objects
  - relative probabilities $p_1, \ldots, p_n$ $(\sum_{i=1}^{n} p_i = 1)$
- Goal: select an object at random
  - the probability of selecting object k is p_k
- Static solution:
  - precompute $\sigma_k = \sum_{i=1}^{k} p_i$
  - generate uniform-$[0, 1)$ random variable x
  - binary search for k s.t. $\sigma_{k-1} \leq x < \sigma_k$



σ_k: $\boxed{0}\boxed{.2}\boxed{.3}\boxed{.6}\boxed{.8}\boxed{1}$

Random object selection problem

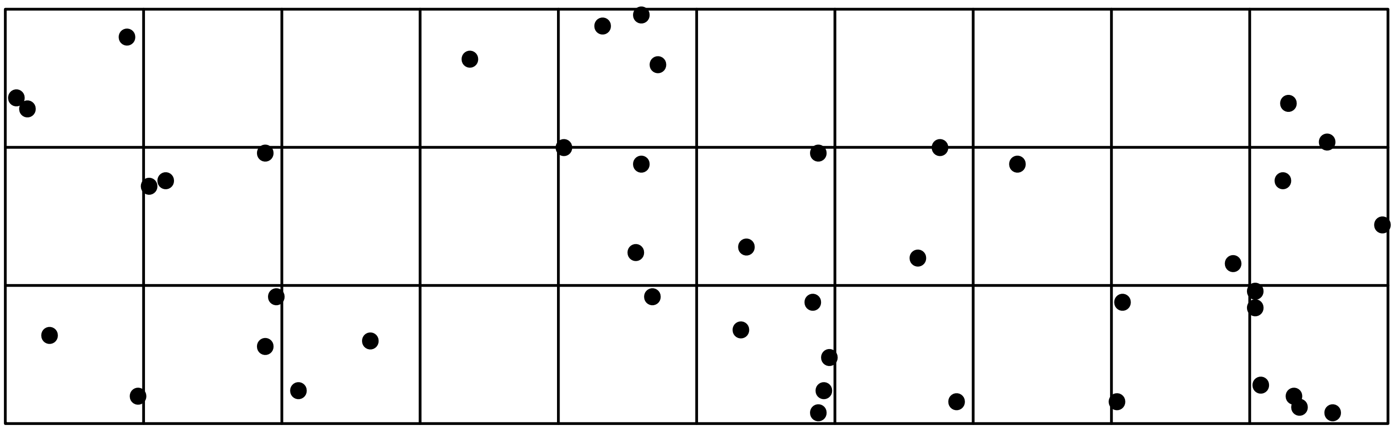
- limitations:
  - probabilities cannot change
  - set of objects cannot change
  - all objects treated equally

Reaction-diffusion equation

- describes how a chemical (or other contaminant) behaves in a fluid
  - chemical movement (e.g. diffusion, fluid flow)
  - increases/decreases in concentration (e.g. reactions, births/deaths)

Simulating

- divide fluid into cells
- concentration $\rightarrow$ discrete number of particles
  - movement of chemical $\rightarrow$ movement of particles between adjacent cells
  - increases/decreases in concentration $\rightarrow$ particle "births" and "deaths"



Simulating

state: collection of all the particle counts

event: e, a particle movement, birth, or death

- modeled as Markov process
- time of next occurrence: exponential random variable with rate λ_e
- function of particle counts, spatial location

Straightforward simulation

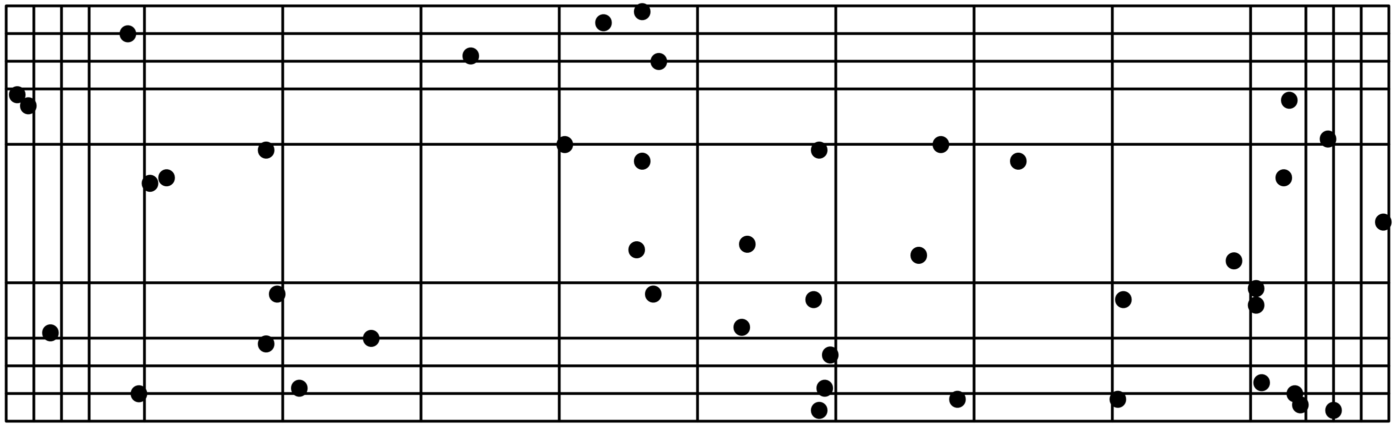
- generate n exponential random variables (time of events)
- find random variable. with smallest value (select event)
- simulate event
- update rates
- re-generate random variables as needed, due to changed rates

More efficient simulation

- let $\lambda_E = \sum_{e \in E} \lambda_e$, sum of all rates
- generate one λ_E-exponential r.v. (time of events)
- select event randomly s.t. event e has probability $p_e = \lambda_e / \lambda_E$
- simulate event
- update rates

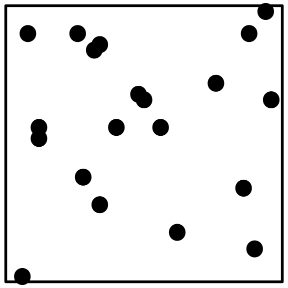
Splitting and merging

- some areas are of greater interest
  - e.g. modeling pollution in a lake → more interested in areas close to shore
  - grid may be non-uniform



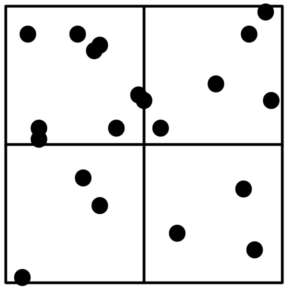
Splitting and merging

- some areas are of greater interest
  - e.g. modeling pollution in a lake → more interested in areas close to shore
  - grid may be non-uniform
- grid may change dynamically
  - e.g. more interested in areas of high concentration
  - split/merge cells



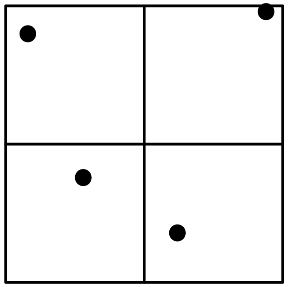
Splitting and merging

- some areas are of greater interest
  - e.g. modeling pollution in a lake → more interested in areas close to shore
  - grid may be non-uniform
- grid may change dynamically
  - e.g. more interested in areas of high concentration
  - split/merge cells



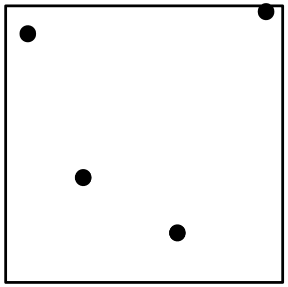
Splitting and merging

- some areas are of greater interest
  - e.g. modeling pollution in a lake $\rightarrow$ more interested in areas close to shore
  - grid may be non-uniform
- grid may change dynamically
  - e.g. more interested in areas of high concentration
  - split/merge cells



Splitting and merging

- some areas are of greater interest
  - e.g. modeling pollution in a lake $\rightarrow$ more interested in areas close to shore
  - grid may be non-uniform
- grid may change dynamically
  - e.g. more interested in areas of high concentration
  - split/merge cells



Problem description

- create a data structure
- Given: set of pairs: events and their rates
  $(e_1, \lambda_1), \ldots, (e_n, \lambda_n)$
- support these operations:

  random select: select an event at random, with probability λ_e/λ_E

  update: set the rate of an event to a new value

  split: split one event $\rightarrow$ a number of events, distribute rate evenly

  merge: merge a number of events $\rightarrow$ one event with rate equal to sum

Problem description

- create a data structure
- Given: set of pairs: events and their rates
  $(e_1, \lambda_1), \ldots, (e_n, \lambda_n)$
- support these operations:

  random select: select an event at random, with probability λ_e / λ_E

  update: set the rate of an event to a new value

  split: split one event $\rightarrow$ a number of events, distribute rate evenly

  merge: merge a number of events $\rightarrow$ one event with rate equal to sum

  insert: add a new event along with its rate

  delete: remove an event along with its rate

Selectable partial sums

- related to simple random object selection problem
- given non-negative *keys* $a_1, \dots, a_n$
- support these operations:

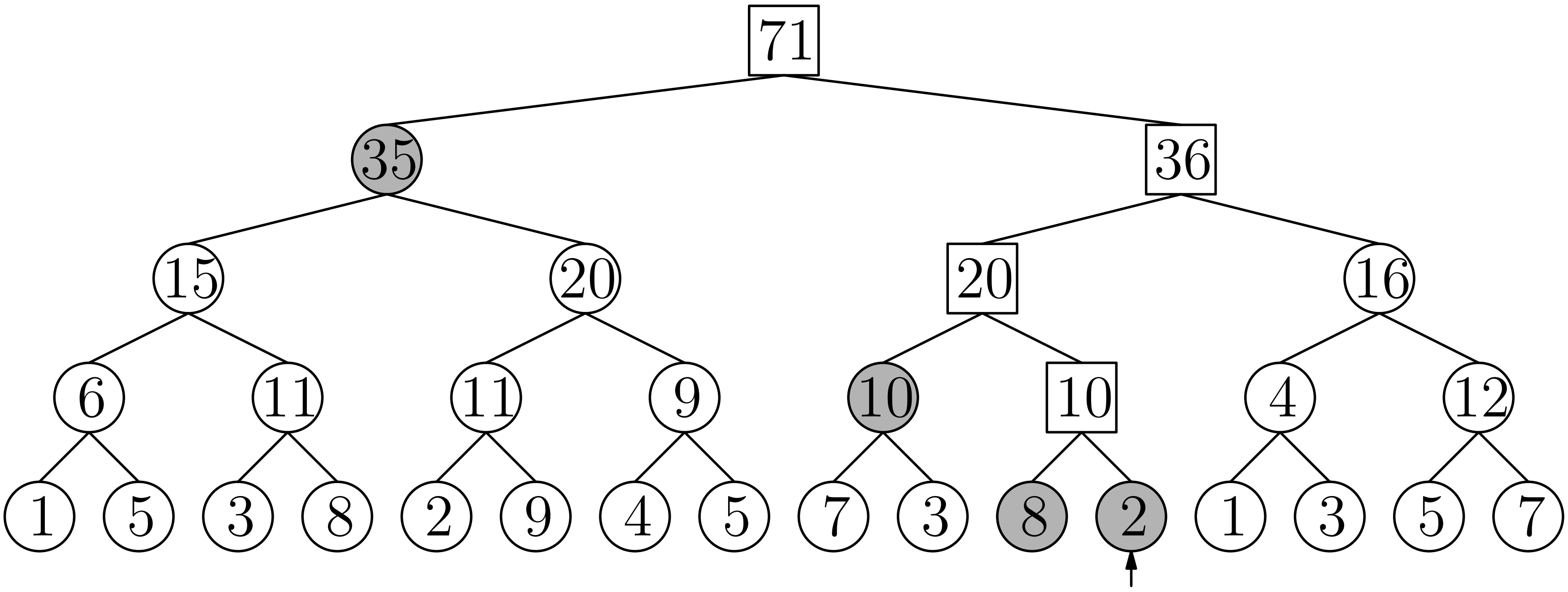
  update: set the value of a key to a new value

  sum: given k, calculate $\sigma_k = \sum_{i=1}^{k} a_i$

  select: given target t, find k such that $\sigma_{k-1} \leq t < \sigma_k$
- to solve random object selection problem:
  - probabilities $p_i \leftrightarrow a_i$
  - uniform-$[0, 1)$ random variable $x \leftrightarrow t$

Binary trees and selectable partial sums

- simplest sub-linear data structure



Previous work on selectable partial sums

Pătraşcu and Demaine: $\Theta(1 + \lg n / \lg(b/\delta))$ on b-bit machine, δ-bit additive changes (upper and lower bounds)

Raman, Raman, Rao: *succinct data structure* $kn + o(kn)$ space, $O(\lg n / \lg \lg n)$ time

Hon, Sadakane, Sung: keys up to $O(\lg \lg n)$ bits, trade off between update and queries

Moffat: $O(\log(1 + k))$ time to update, sum, or select kth key a_k

Hampapuram and Fredman: updates and sums have different probabilities, but does not support selections

Optimal search trees/coding systems

- use a tree to solve selectable partial sums
- consider different access probabilities
- optimal search trees: minimize expected cost for a search
  - entropy $H = \sum_{i=1}^{n} p_i \log 1/p_i$
  - expected search time: between $H - \log H - \log e + 1$ and $H + 2$
- some coding systems (e.g. Huffman): correspondence with trees
  - also tries to minimize expected access cost

Previous work on coding

Faller, Gallager, Knuth: dynamic Huffman coding

Vitter: improved FGK algorithm

Gagie: dynamic Shannon coding

Research goals

- support the operations with the following running times:
  select: $O(\log 1/p)$
  update: $O(\log 1/p)$, or maybe $O(f(\Delta\lambda) \log 1/p)$
  split: ?
  merge: ?
  insert: $O(\log 1/p)$
  delete: $O(\log 1/p)$

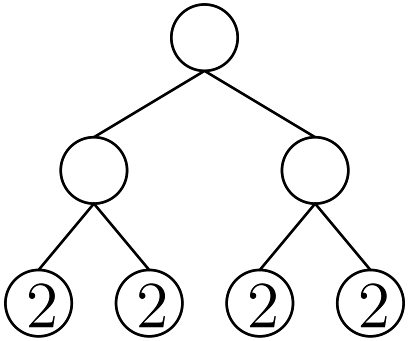
Splitting

- under a standard tree: trivial

  ⑧

Splitting

- under a standard tree: trivial

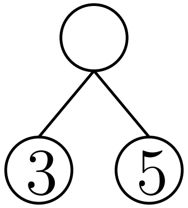


Splitting

- under a standard tree: trivial
- harder if tree is stored in special structure (e.g. Vitter), or data structure has other constraints

Merging

- harder than splitting
- if nodes to be merged have common parent → easy



Merging

- harder than splitting
- if nodes to be merged have common parent → easy

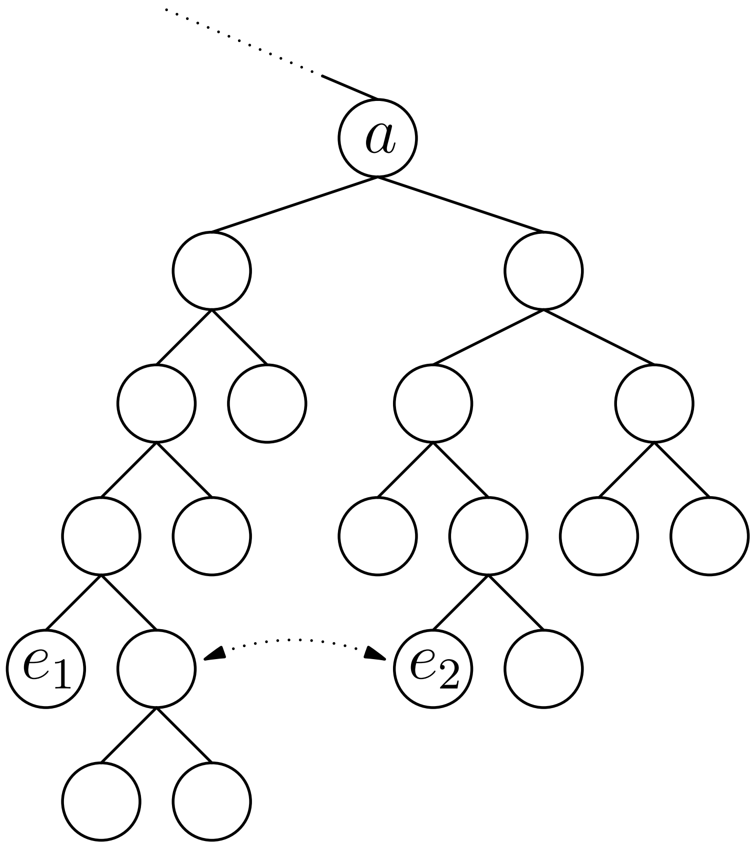
⑧

Merging

- harder than splitting
- if nodes to be merged have common parent $\rightarrow$ easy
- otherwise, need to remove old nodes, insert new node $\rightarrow$ minimize time

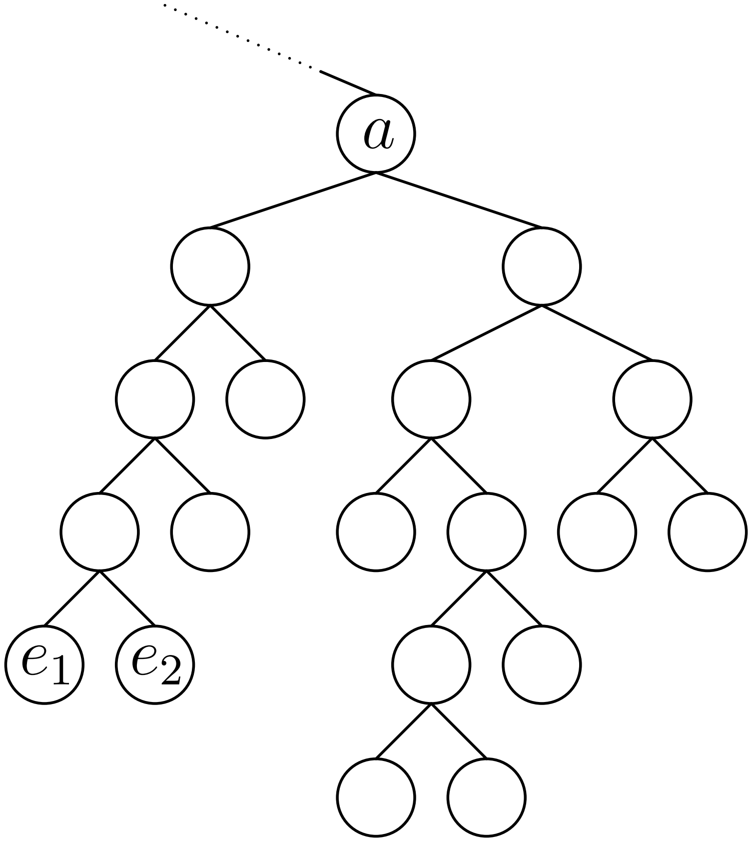
Merging

- if events on same level:



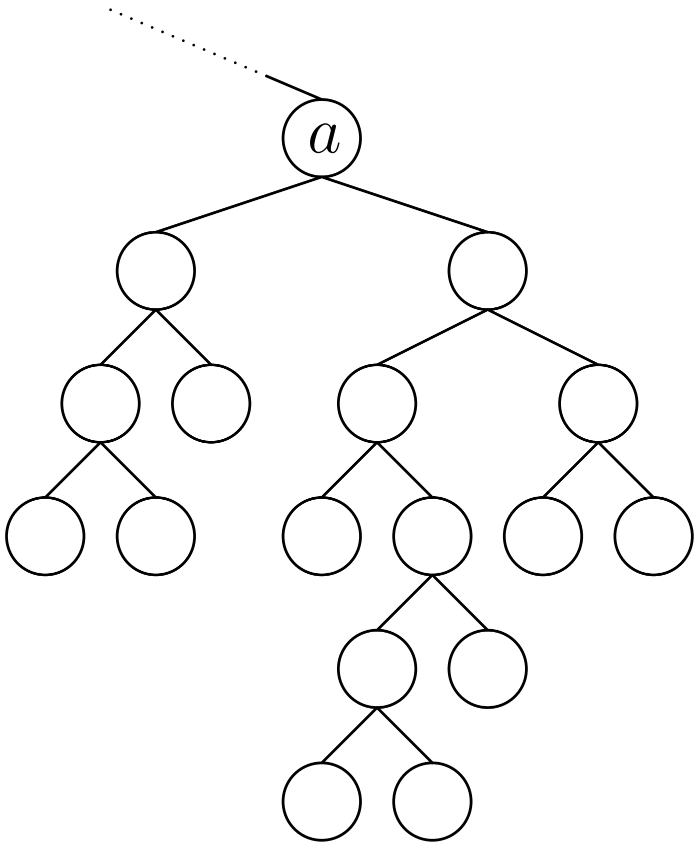
Merging

- if events on same level:



Merging

- if events on same level:

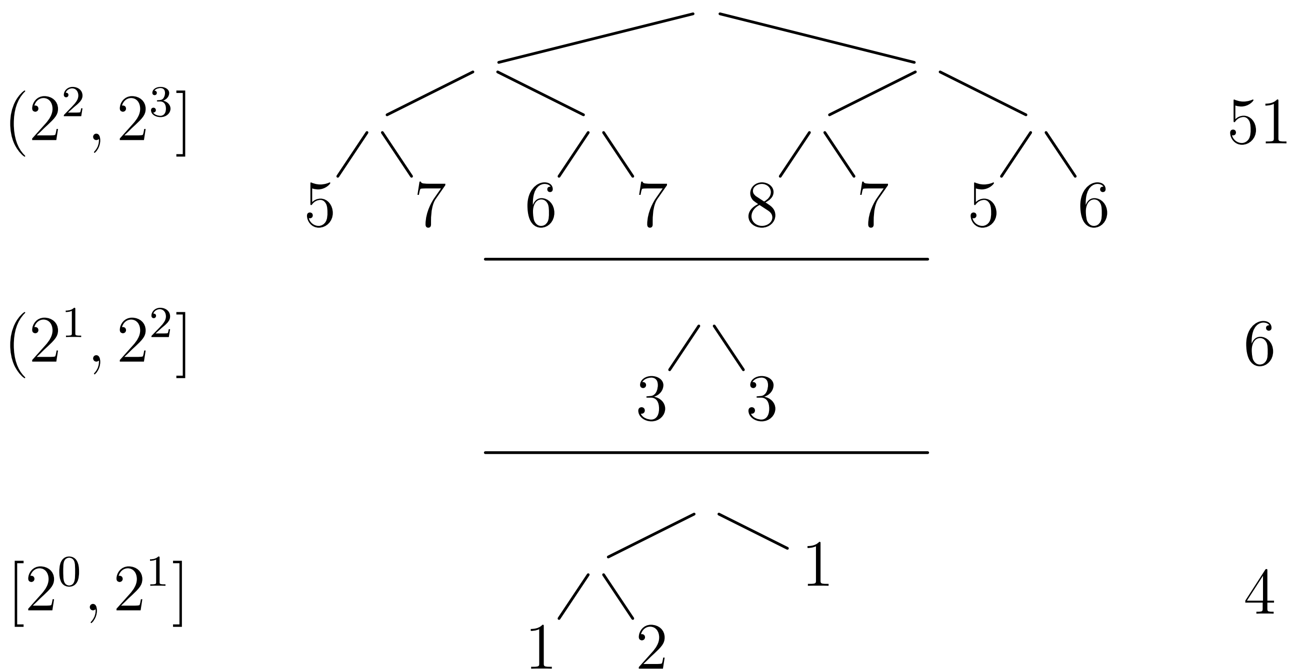


Approaches

- try to adapt previous work
  - Hampapuram and Fredman: consider different probabilities, but do not support needed operations
  - coding systems
    - support updating weights, but only increment/decrement by one $\rightarrow O(\Delta\lambda \log 1/p)$
    - also can support insert/delete, but too slow

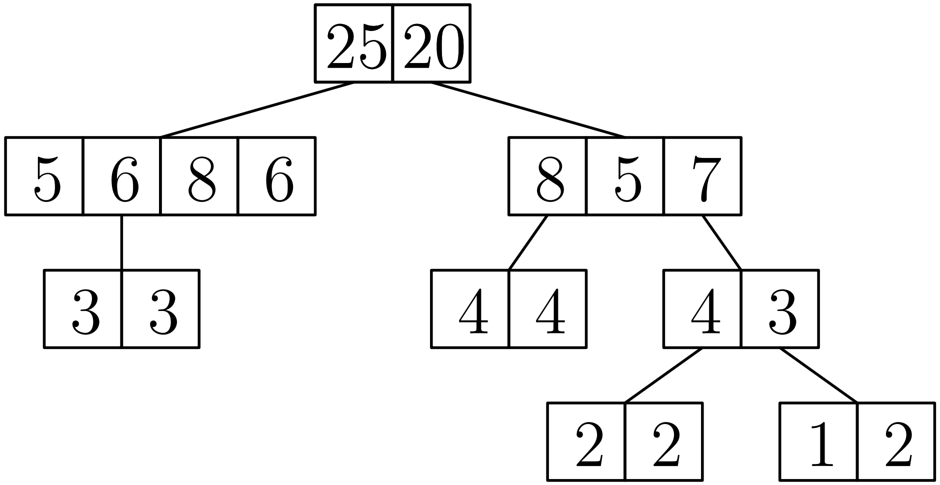
Approaches

- group nodes according to rate
- put each group in a balanced tree



Approaches

- B-tree-like structure
- elements in each node are within a range of rates



Approaches

- divide space into cells
- similar to quad trees
- need further research

Summary

- random object selection problem with split and merge
- speed up simulation of reaction-diffusion equation
- related to partial sums, optimal search trees, coding systems
  - starting point in our research
- other approaches based on
  - grouping nodes by rates
  - B-trees
- quad trees