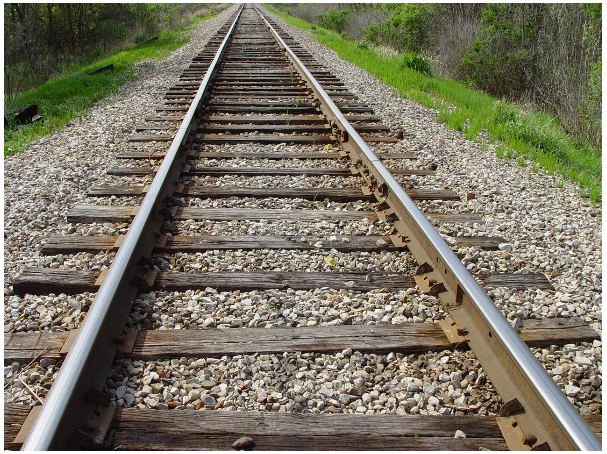
Karlsruhe Institute of Technology

Train Tracks with Gaps
Applying the Probabilistic Method to Trains

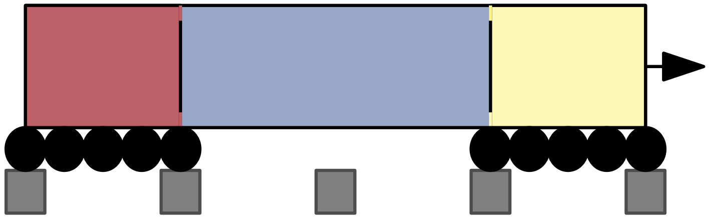
Seminar Algorithmentechnik · November 10, 2023
Robert Krause

INSTITUTE OF THEORETICAL INFORMATICS · ALGORITHMICS GROUP

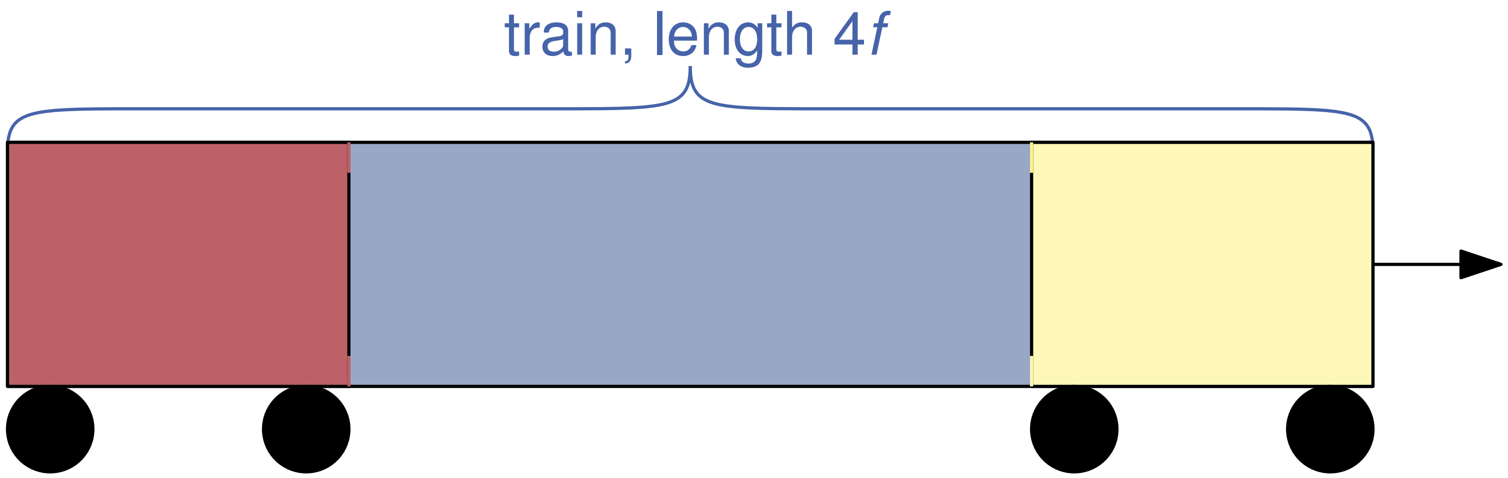
https://www.railway-technology.com/projects/amtraks-airo-passenger-train-usa/



https://www.pics4learning.com/details.php?img=t

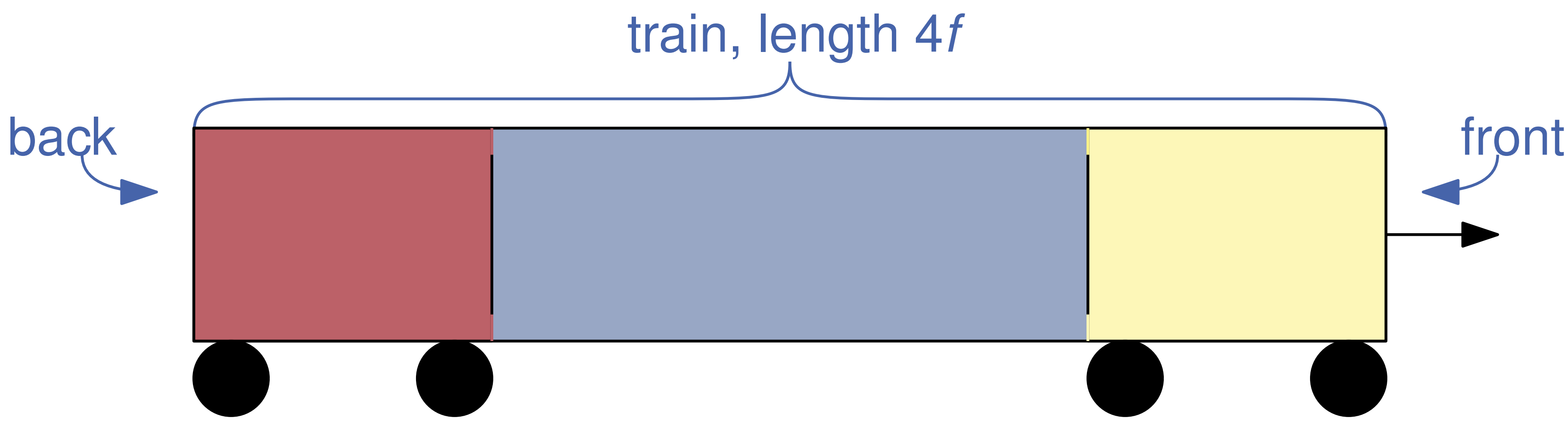


Tains, Tracks and Gaps

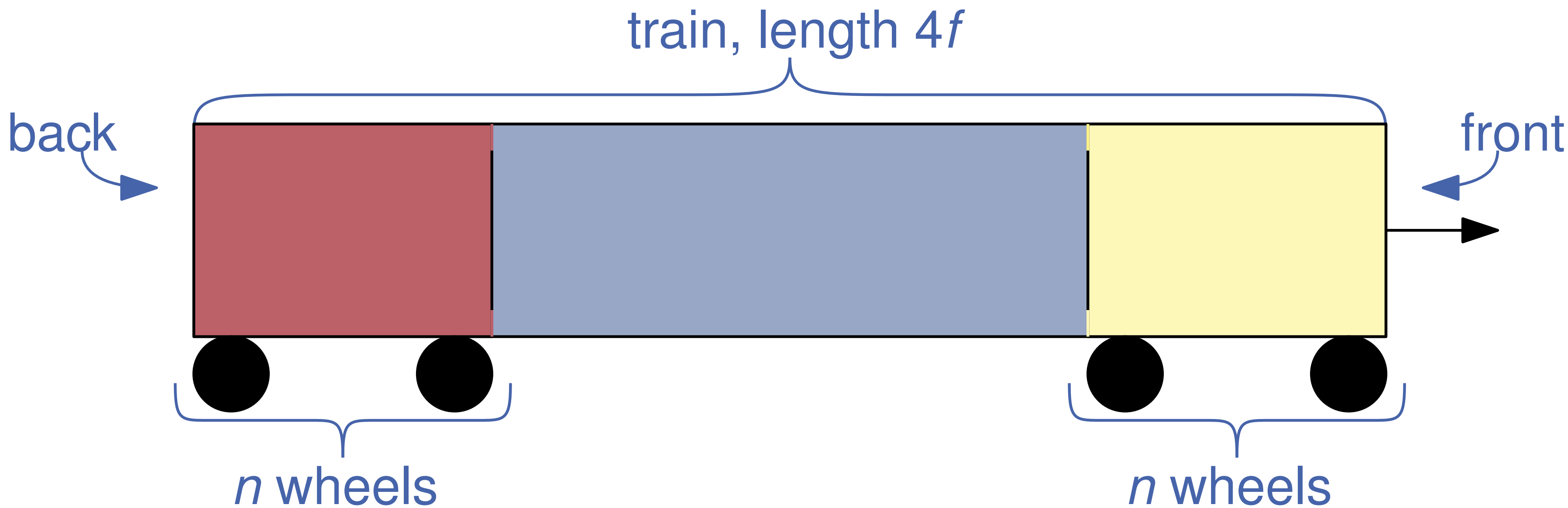


Institute of Theoretical Informatics
Algorithmics Group

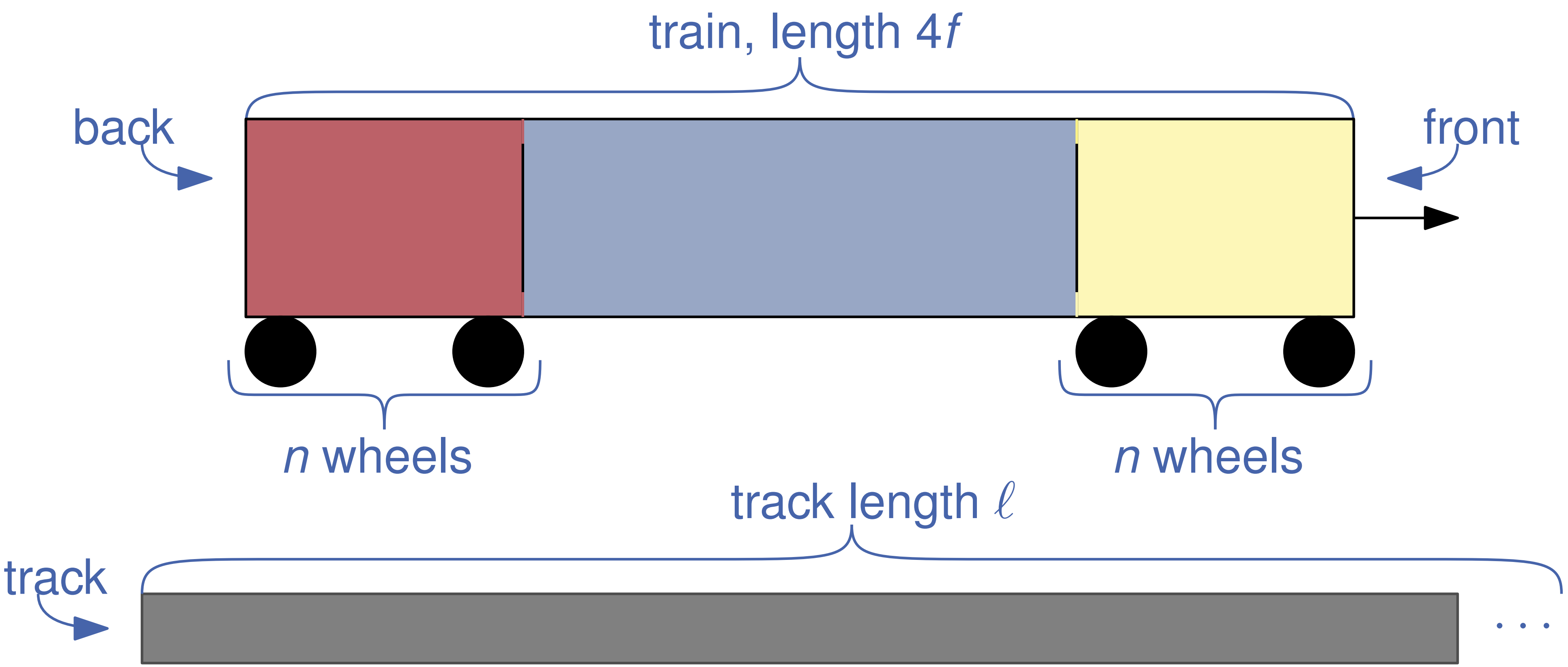
Tains, Tracks and Gaps



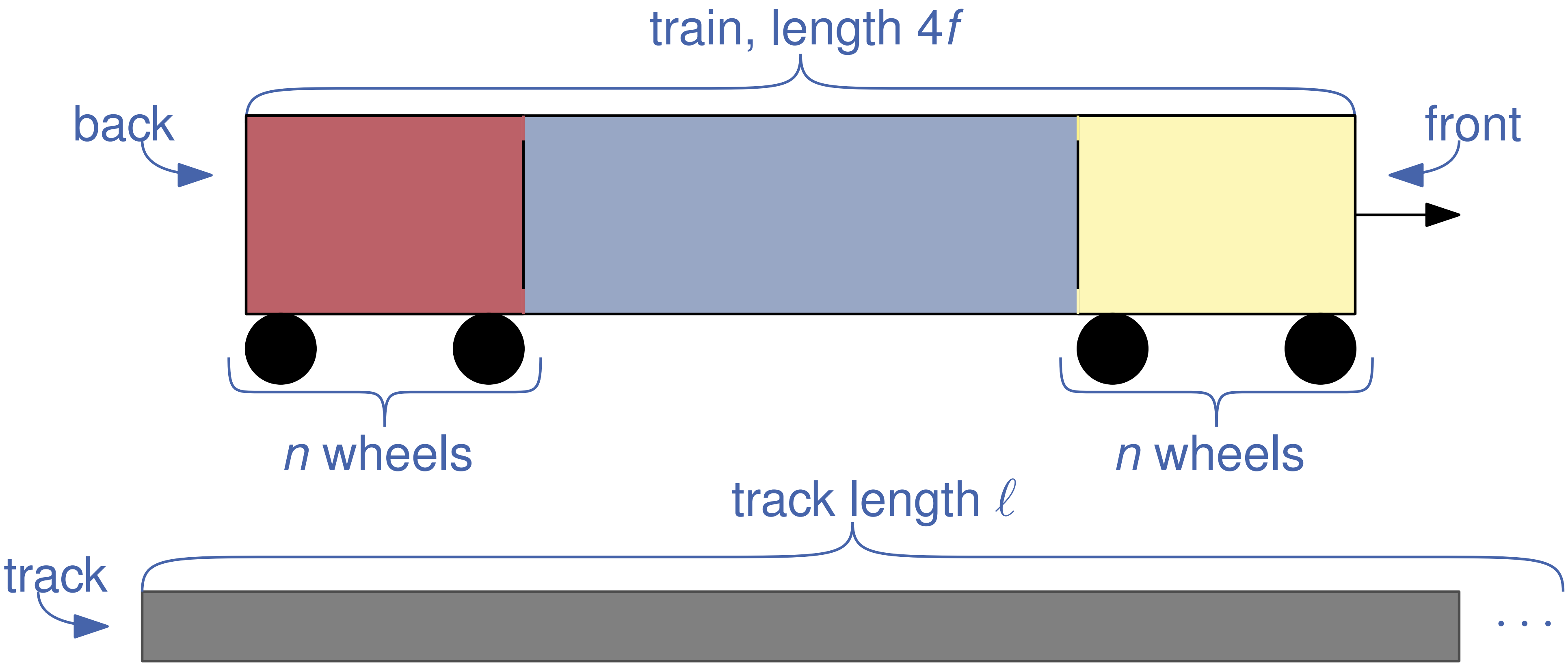
Tains, Tracks and Gaps



Tains, Tracks and Gaps



Tains, Tracks and Gaps



- trains drive on uninterrupted tracks
- very wasteful

$\Rightarrow$ build as little track as neccessary

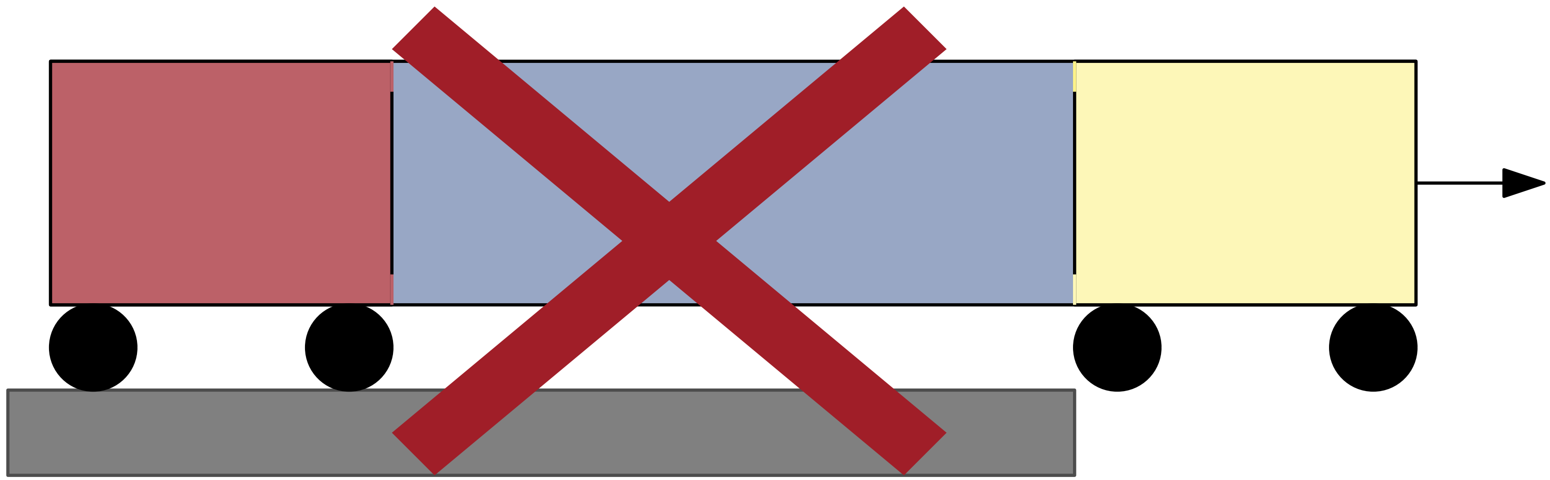
Institute of Theoretical Informatics
Algorithmics Group

How much track do we really need?

- front and back quarter have to always be supported

Institute of Theoretical Informatics
Algorithmics Group

How much track do we really need?



- front and back quarter have to always be supported

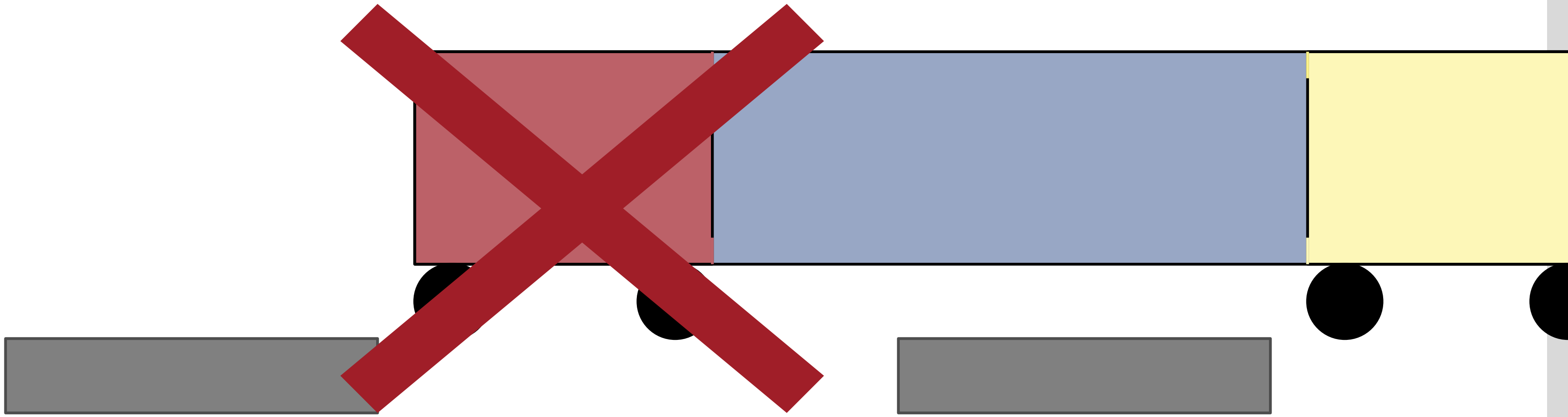
Institute of Theoretical Informatics
Algorithmics Group

How much track do we really need?

- front and back quarter have to always be supported

Institute of Theoretical Informatics
Algorithmics Group

How much track do we really need?



- front and back quarter have to always be supported

Institute of Theoretical Informatics
Algorithmics Group

How much track do we really need?

- front and back quarter have to always be supported
- we will show:

Institute of Theoretical Informatics
Algorithmics Group

How much track do we really need?

- front and back quarter have to always be supported
- we will show:
  - $\mathcal{O}(\ell/n)$ track for equally spaced wheels
  - $\mathcal{O}(\frac{\ell \ln n}{n})$ track for arbitrary wheel arrangements

Institute of Theoretical Informatics
Algorithmics Group

Probabilistic Method

- method for proving the existence of a mathematical object
- choose objects randomly, if the probability for prescribed object is greater 0 then it must exist

Institute of Theoretical Informatics
Algorithmics Group

Probabilistic Method

- method for proving the existence of a mathematical object
- choose objects randomly, if the probability for prescribed object is greater 0 then it must exist

Lemma: It is possible to flip a coin three times so that the number of tails is at least 2

Probabilistic Method

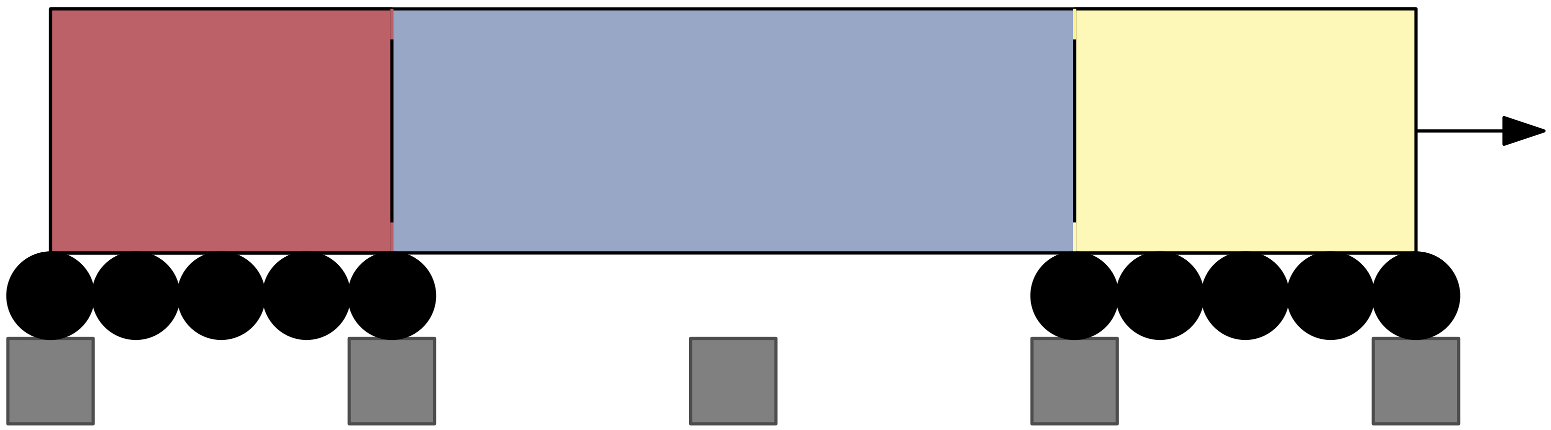
- method for proving the existence of a mathematical object
- choose objects randomly, if the probability for prescribed object is greater 0 then it must exist

Lemma: It is possible to flip a coin three times so that the number of tails is at least 2

Proof:

- expected value is 1.5
- outcome is integer
- there exists an outcome $\geq 1.5 \Rightarrow$ there have to be at least 2

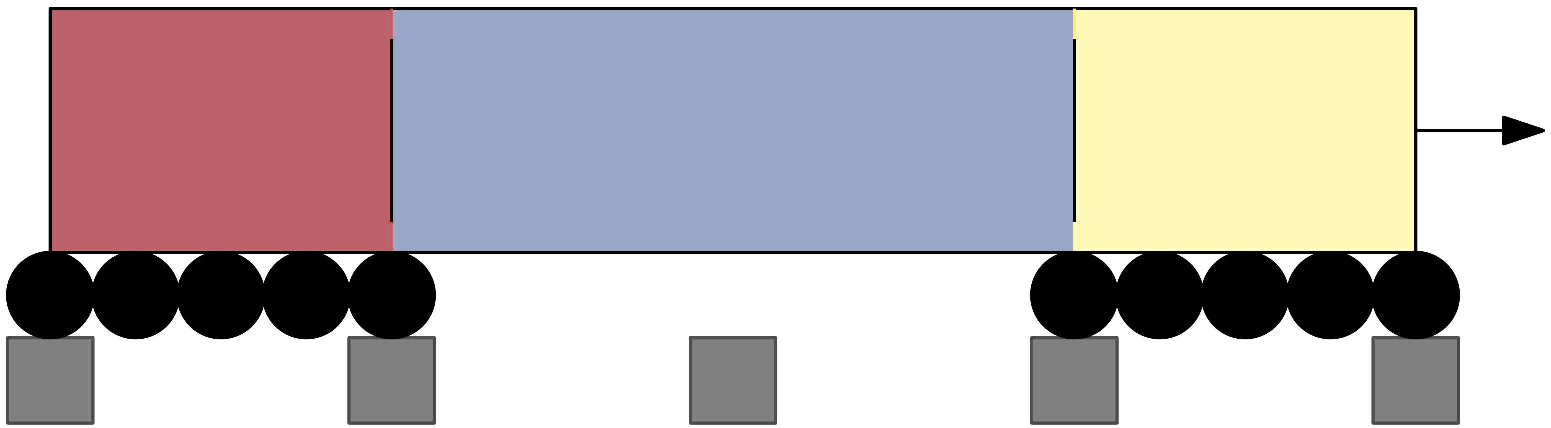
Evenly spaced wheels: Lower Bound



Placing a track of length $\frac{1}{4n}$ every quarter of a train length is optimal

Institute of Theoretical Informatics
Algorithmics Group

Evenly spaced wheels: Lower Bound

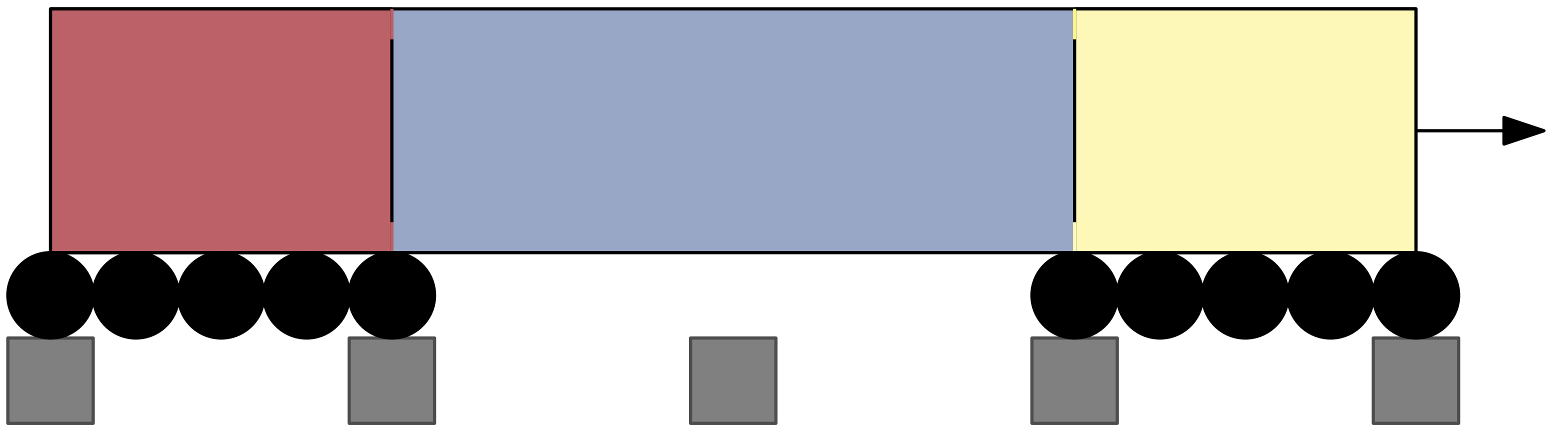


Placing a track of length $\frac{1}{4n}$ every quarter of a train length is optimal

- assume a portion $< \frac{1}{n}$ of the track is built
- probability w_i for wheel i to be supported by track is $< \frac{1}{n}$

Institute of Theoretical Informatics
Algorithmics Group

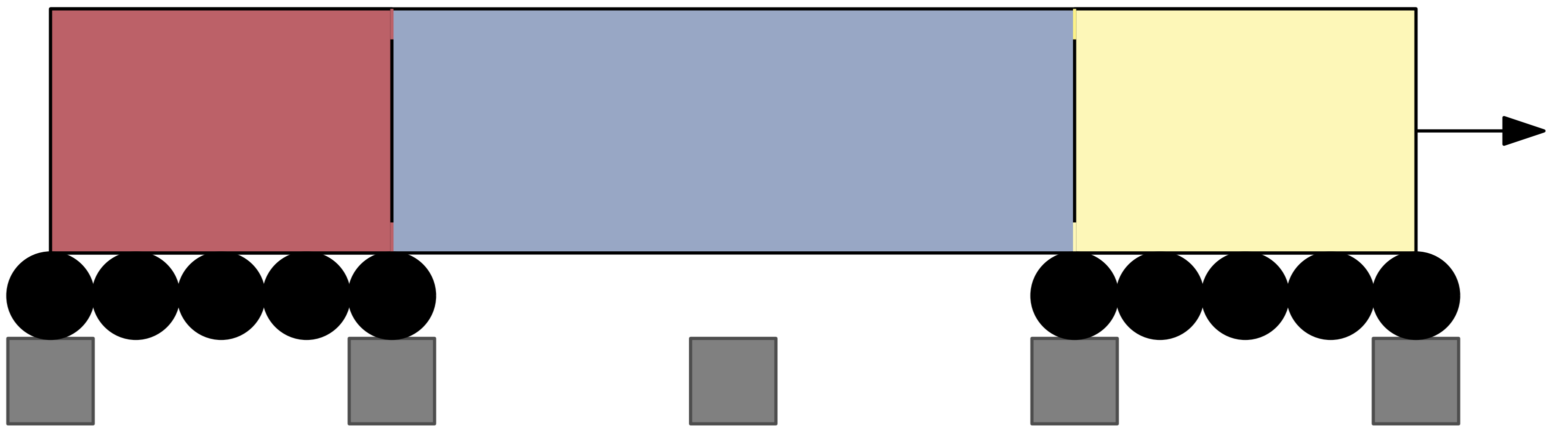
Evenly spaced wheels: Lower Bound



Placing a track of length $\frac{1}{4n}$ every quarter of a train length is optimal

- assume a portion $< \frac{1}{n}$ of the track is built
- probability w_i for wheel i to be supported by track is $< \frac{1}{n}$
- only looking at wheels in the back
- using union bound: $\mathbb{P}(\bigcup_{i=1}^{n} w_i) \leq \sum_{i=1}^{n} \mathbb{P}(w_i) < \sum_{i=1}^{n} \frac{1}{n} = 1$

Evenly spaced wheels: Lower Bound

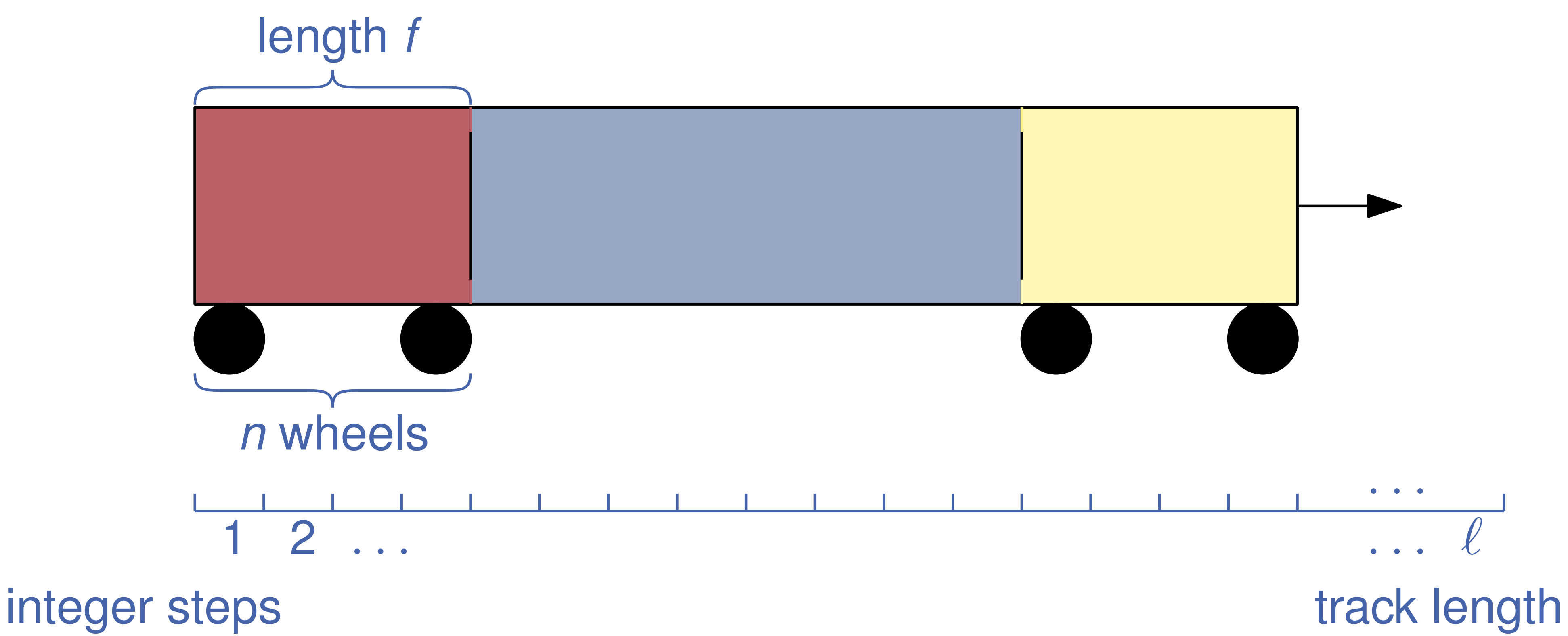


Placing a track of length $\frac{1}{4n}$ every quarter of a train length is optimal

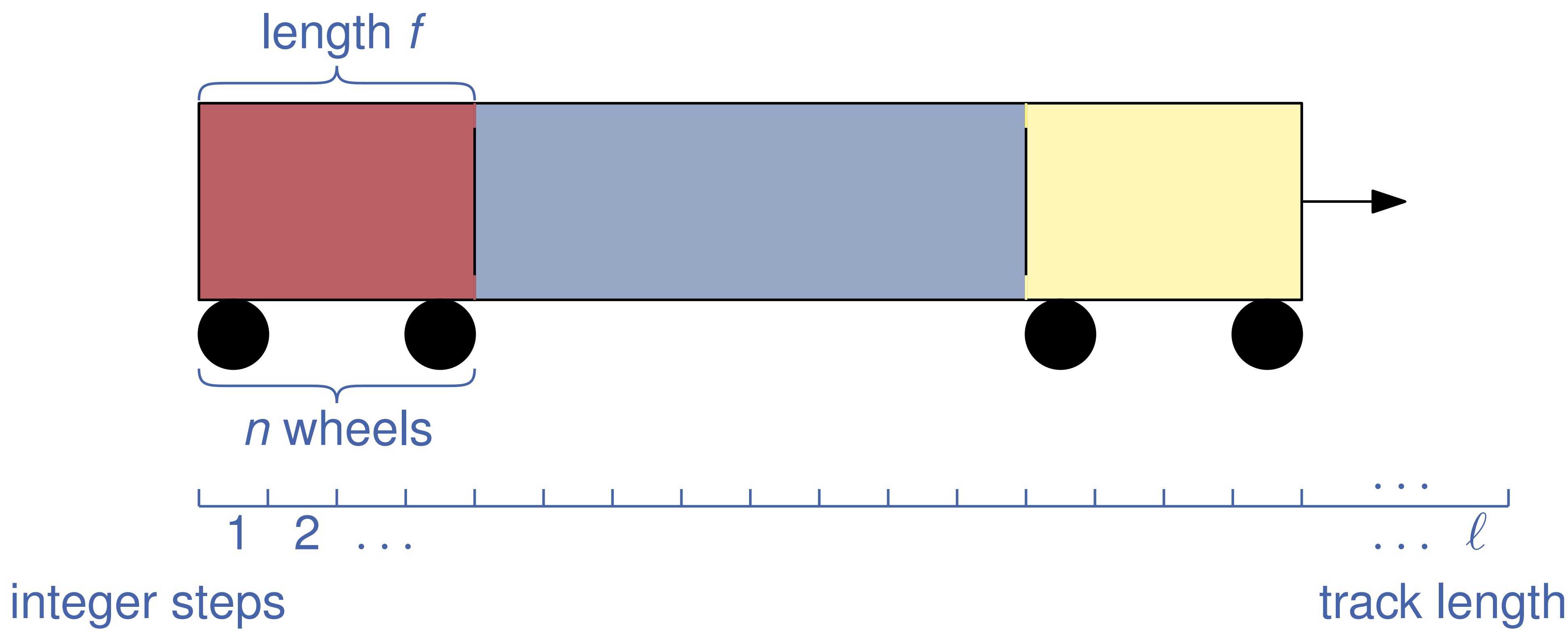
- assume a portion $< \frac{1}{n}$ of the track is built
- probability w_i for wheel i to be supported by track is $< \frac{1}{n}$
- only looking at wheels in the back
- using union bound: $\mathbb{P}(\bigcup_{i=1}^{n} w_i) \leq \sum_{i=1}^{n} \mathbb{P}(w_i) < \sum_{i=1}^{n} \frac{1}{n} = 1$

$\Rightarrow$ there exists a position where the train falls though

The Setting formalised

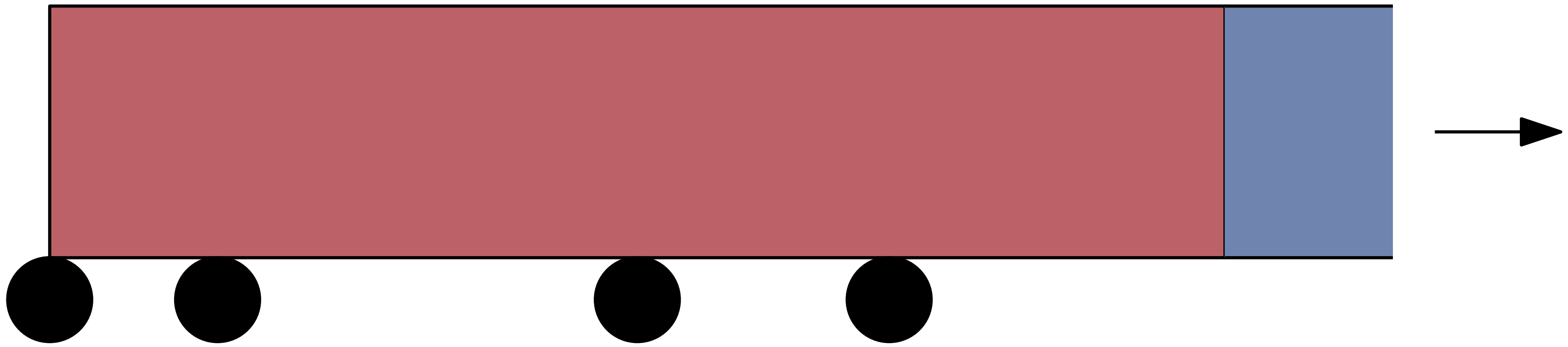


The Setting formalised

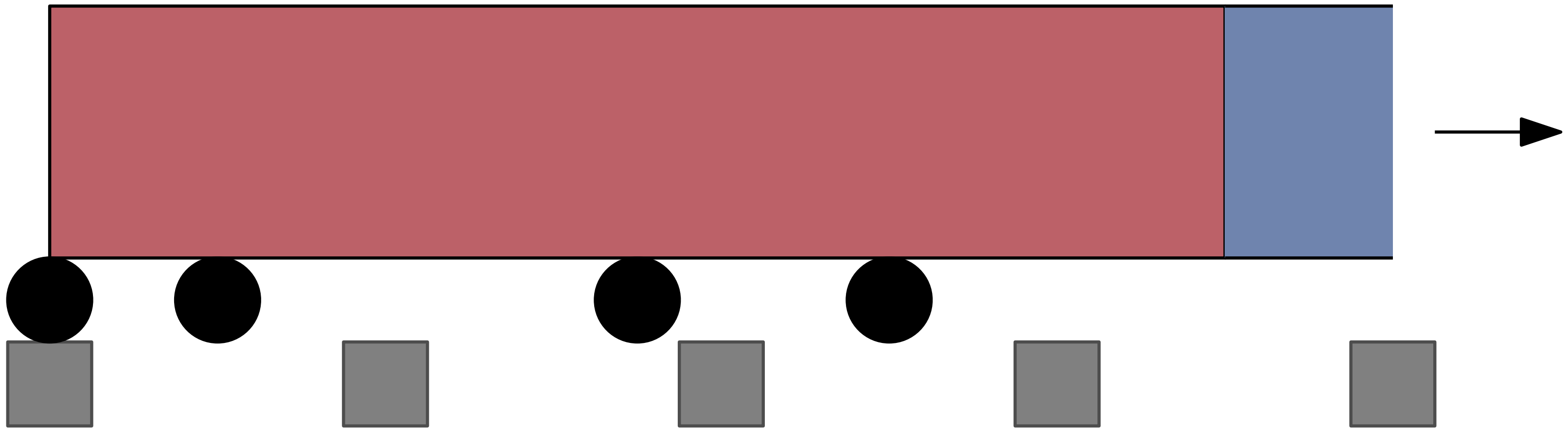


- C = set of wheel positions, from the rear quarter, of size n
- $T \subseteq \{1, 2, \ldots, \ell\}$ = set of pillars
- $C + r = \{c + r \mid c \in C\}$
- T is valid $\Leftrightarrow (C + k) \cap T \neq \emptyset$, for $k = 1, \ldots, \ell - f$

Arbitrary Wheel Arrangements



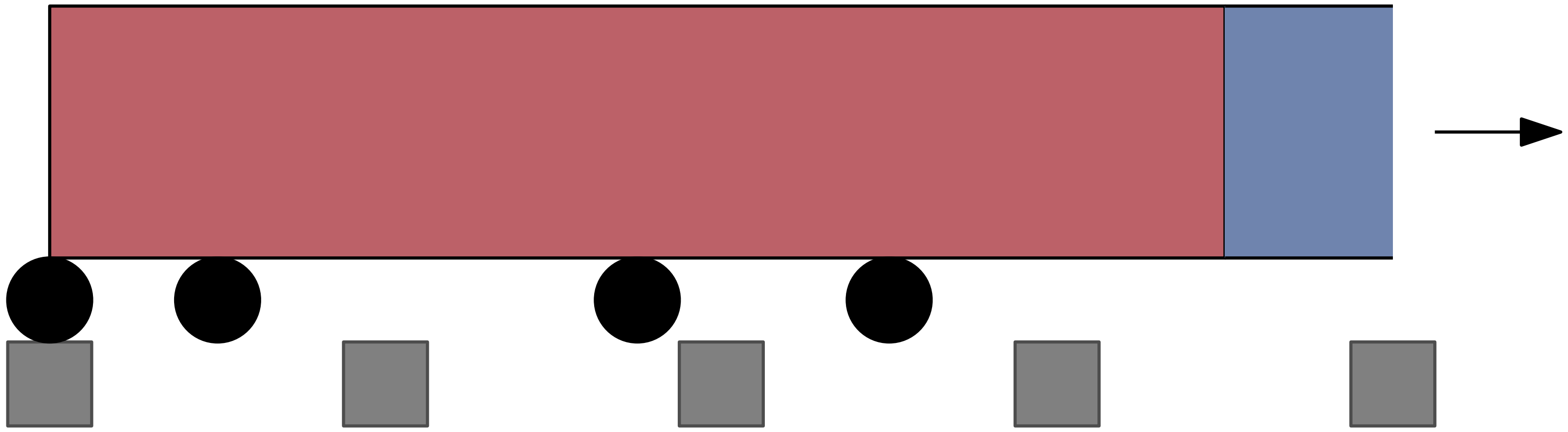
Arbitrary Wheel Arrangements



- every pillar is built with probability $\frac{\ln n}{n}$

Institute of Theoretical Informatics
Algorithmics Group

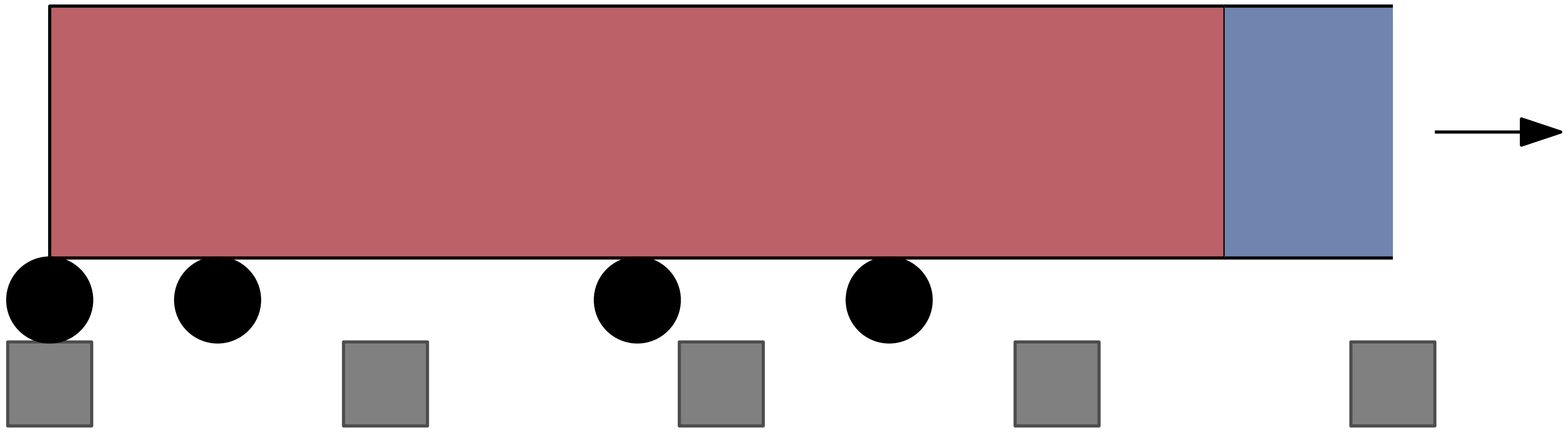
Arbitrary Wheel Arrangements



- every pillar is built with probability $\frac{\ln n}{n}$
- train falls with probabilty $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- out of all the possible train placements only $\frac{1}{n}$-th is problematic

Arbitrary Wheel Arrangements

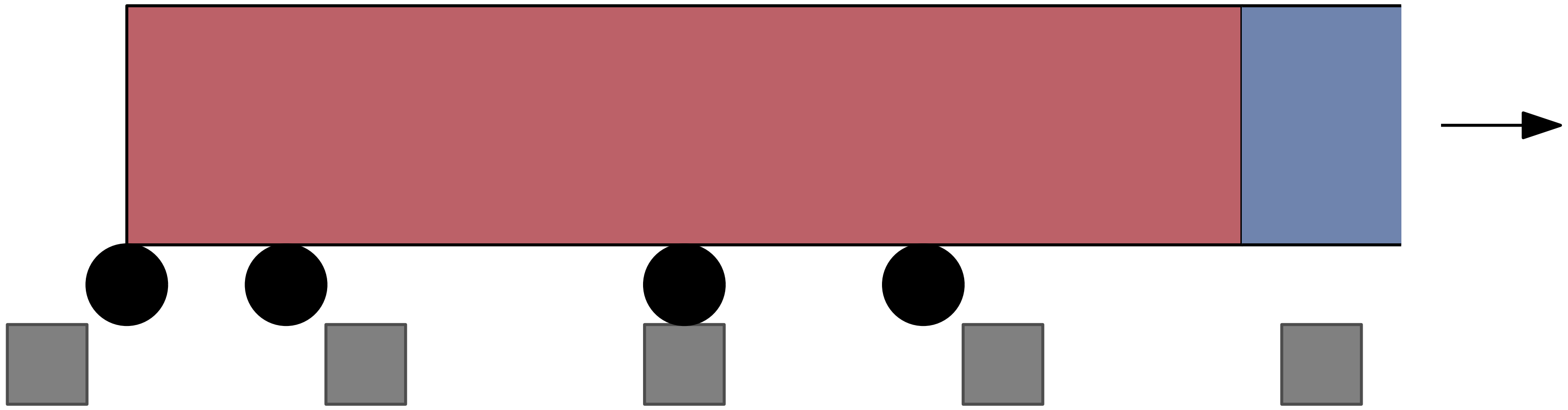
Karlsruhe Institute of Technology



- every pillar is built with probability $\frac{\ln n}{n}$
- train falls with probabilty $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- out of all the possible train placements only $\frac{1}{n}$-th is problematic
- fix these instances by adding a pillar to support the train

Institute of Theoretical Informatics
Algorithmics Group

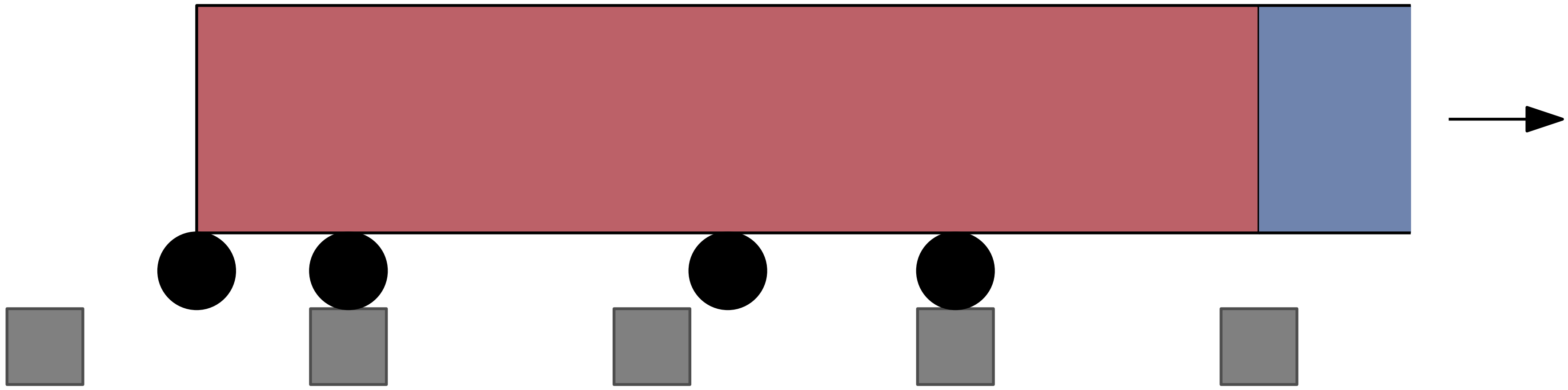
Arbitrary Wheel Arrangements



- every pillar is built with probability $\frac{\ln n}{n}$
- train falls with probabilty $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- out of all the possible train placements only $\frac{1}{n}$-th is problematic
- fix these instances by adding a pillar to support the train

Institute of Theoretical Informatics
Algorithmics Group

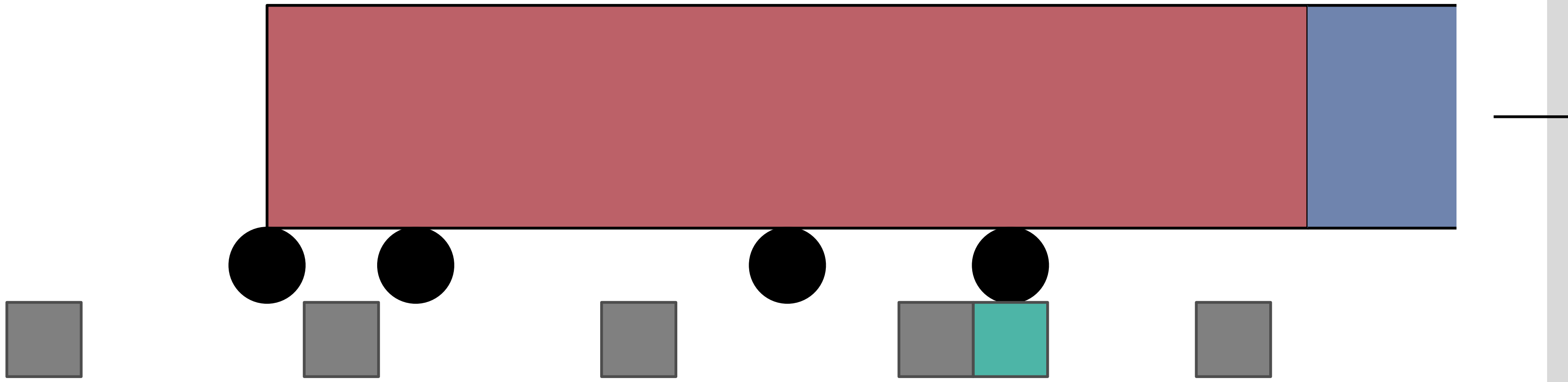
Arbitrary Wheel Arrangements



- every pillar is built with probability $\frac{\ln n}{n}$
- train falls with probabilty $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- out of all the possible train placements only $\frac{1}{n}$-th is problematic
- fix these instances by adding a pillar to support the train

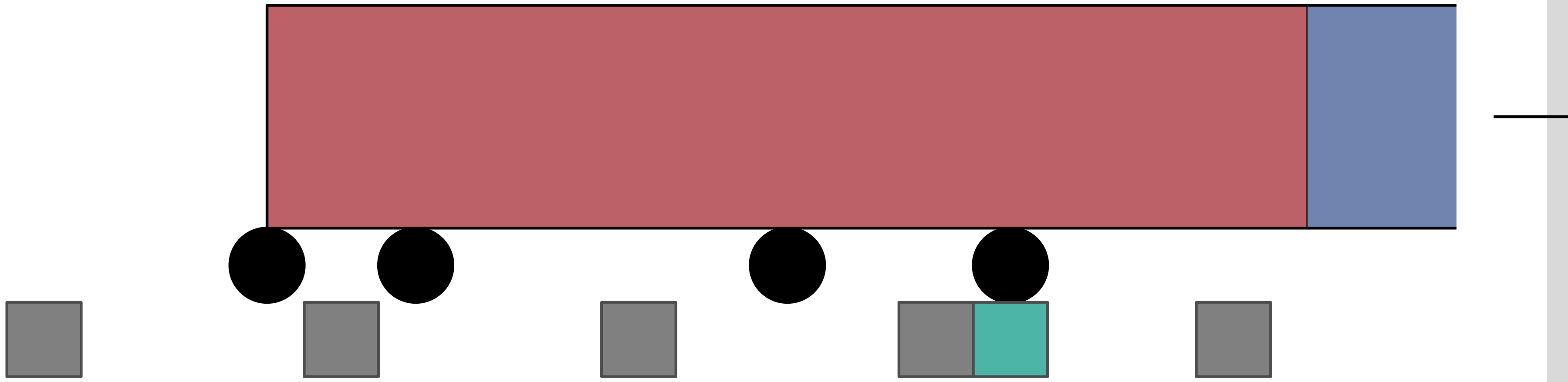
Institute of Theoretical Informatics
Algorithmics Group

Arbitrary Wheel Arrangements



- every pillar is built with probability $\frac{\ln n}{n}$
- train falls with probabilty $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- out of all the possible train placements only $\frac{1}{n}$-th is problematic
- fix these instances by adding a pillar to support the train

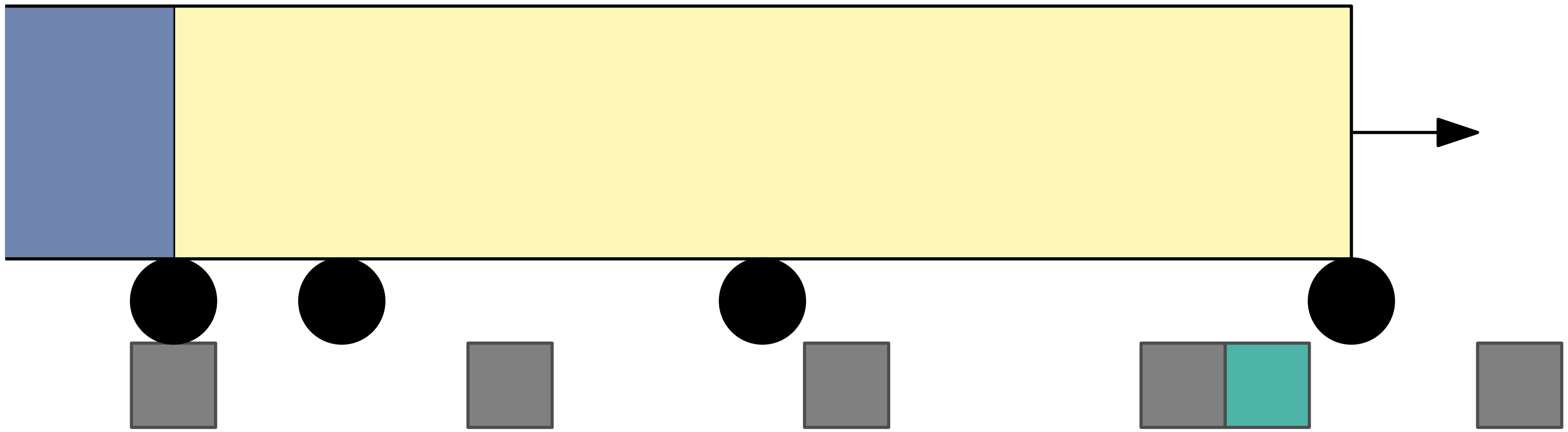
Arbitrary Wheel Arrangements



- every pillar is built with probability $\frac{\ln n}{n}$
- train falls with probabilty $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- out of all the possible train placements only $\frac{1}{n}$-th is problematic
- fix these instances by adding a pillar to support the train
- $\frac{\ln n}{n} \cdot \ell + \frac{1}{n} \cdot \ell = (\frac{1+\ln n}{n}) \cdot \ell$ exp. tracks
- $\mathcal{O}(\ell n)$ runtime

Why is it enough to look at the rear quarter?

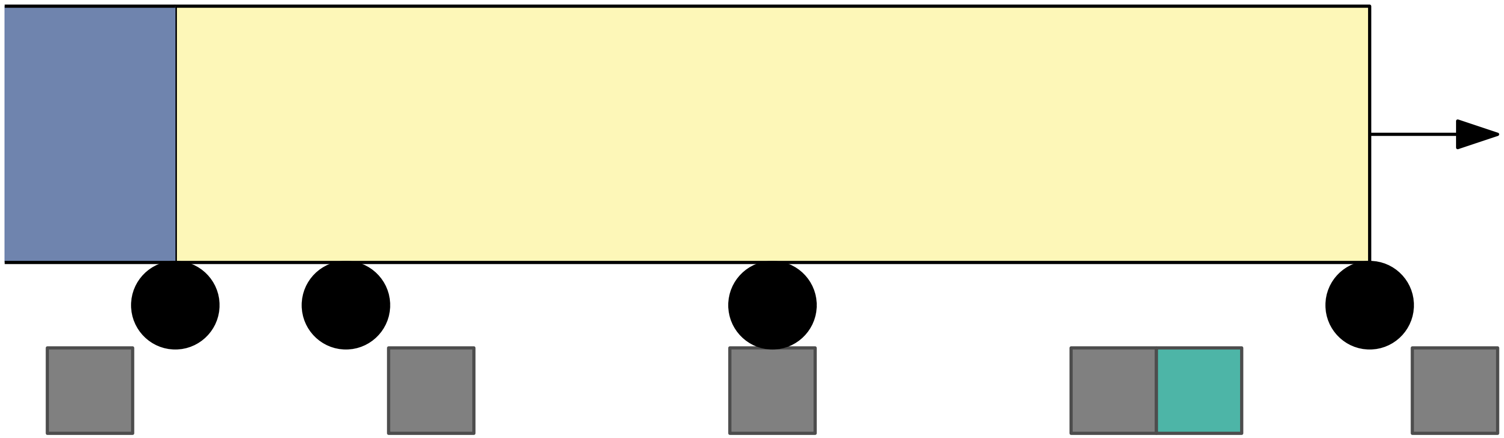
Karlsruhe Institute of Technology



- keep the calculated tracks
- check front quarter for every position
- running time still $\mathcal{O}(\ell n)$

Institute of Theoretical Informatics
Algorithmics Group

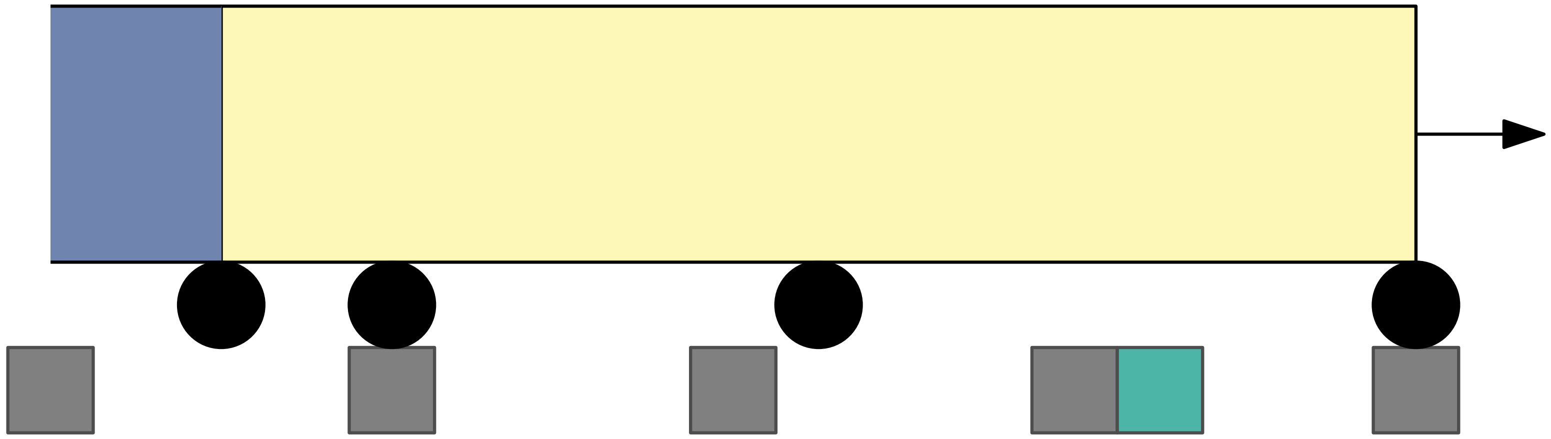
Why is it enough to look at the rear quarter?



- keep the calculated tracks
- check front quarter for every position
- running time still $\mathcal{O}(\ell n)$

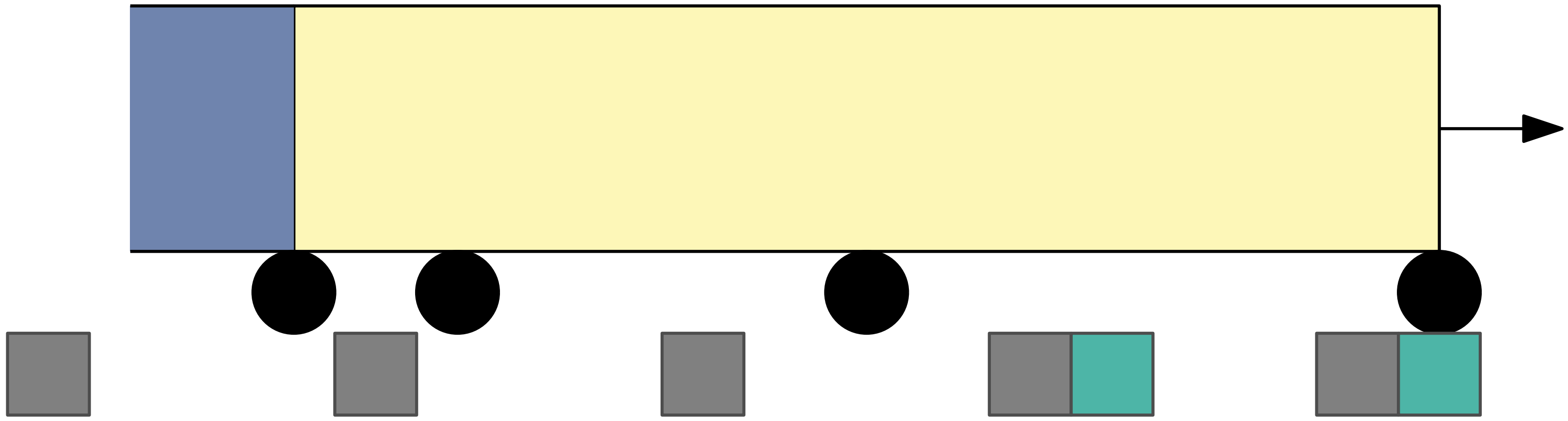
Institute of Theoretical Informatics
Algorithmics Group

Why is it enough to look at the rear quarter?



- keep the calculated tracks
- check front quarter for every position
- running time still $\mathcal{O}(\ell n)$

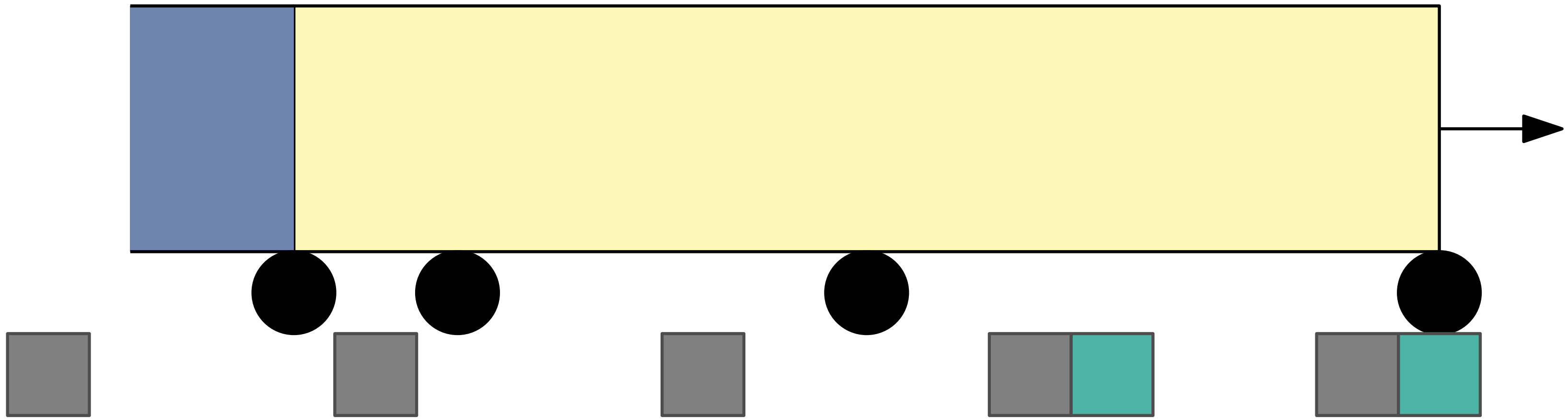
Why is it enough to look at the rear quarter?



- keep the calculated tracks
- check front quarter for every position
- running time still $\mathcal{O}(\ell n)$

Institute of Theoretical Informatics
Algorithmics Group

Why is it enough to look at the rear quarter?



- keep the calculated tracks
- check front quarter for every position
- running time still $\mathcal{O}(\ell n)$
- train falls with probabilty $(1 - \frac{1+\ln n}{n})^n \leq \frac{1}{e^{1+\ln n}} = \frac{1}{ne}$
- build additional exp. $\frac{\ell}{ne}$ tracks

Institute of Theoretical Informatics
Algorithmics Group

Further Algorithms

- what we have seen so far:

	lower-bound	upper-bound
even	$\frac{\ell}{n}$	$\frac{\ell}{n}$
arbitrary	$(\frac{\ell \ln n}{n})$	$\frac{\ell \ln n}{n}$

Institute of Theoretical Informatics
Algorithmics Group

Further Algorithms

- what we have seen so far:

	lower-bound	upper-bound
even	$\frac{\ell}{n}$	$\frac{\ell}{n}$
arbitrary	$(\frac{\ell \ln n}{n})$	$\frac{\ell \ln n}{n}$

- Goals for algorithms:
  1. correctness: Find set T such that for each $k \in \{0, 1, \ldots, \ell - f\}$, the set $(C + k) \cap T \neq \emptyset$
  2. runtime: exp. $\mathcal{O}(n\ell)$
  3. track length: $|T| \in$ exp. $\mathcal{O}(\frac{\ell \ln n}{n})$

Institute of Theoretical Informatics
Algorithmics Group

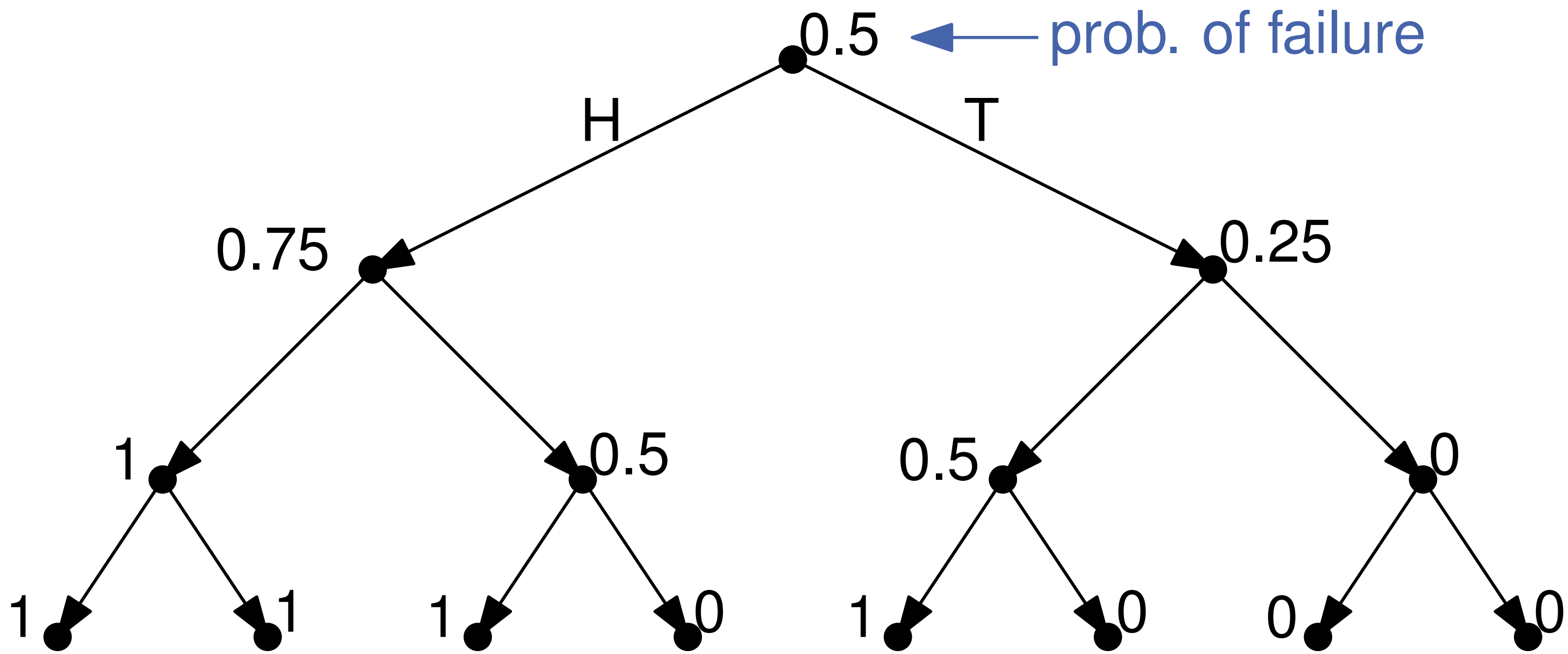
Method of Conditional Probabilities

- Convert proof via the probabilistic method to efficient, deterministic alogrithm

Institute of Theoretical Informatics
Algorithmics Group

Method of Conditional Probabilities

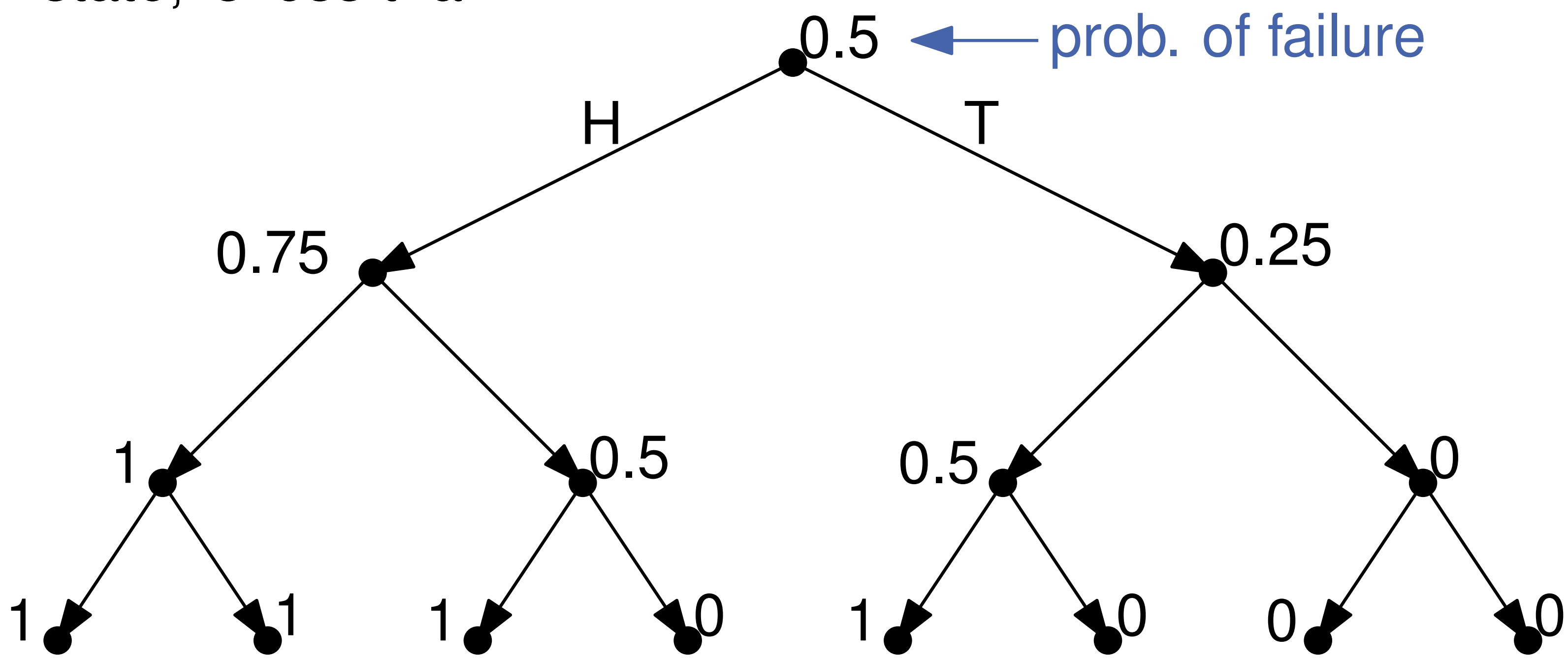
- Convert proof via the probabilistic method to efficient, deterministic alogrithm
- replace random root-to-leaf walk with deterministic walk



Institute of Theoretical Informatics
Algorithmics Group

Method of Conditional Probabilities

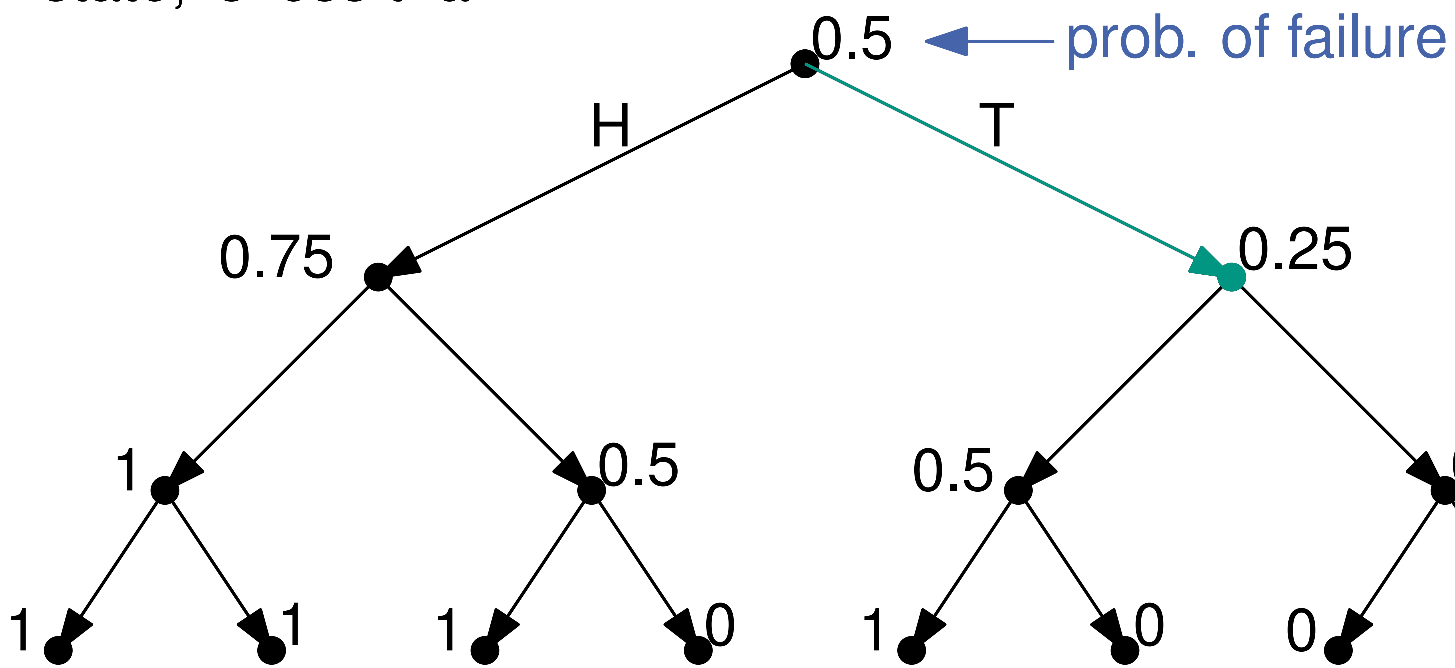
- Convert proof via the probabilistic method to efficient, deterministic alogrithm
- replace random root-to-leaf walk with deterministic walk
- maintain invariant: the conditional probability of failure, given the current state, is less than 1



Institute of Theoretical Informatics
Algorithmics Group

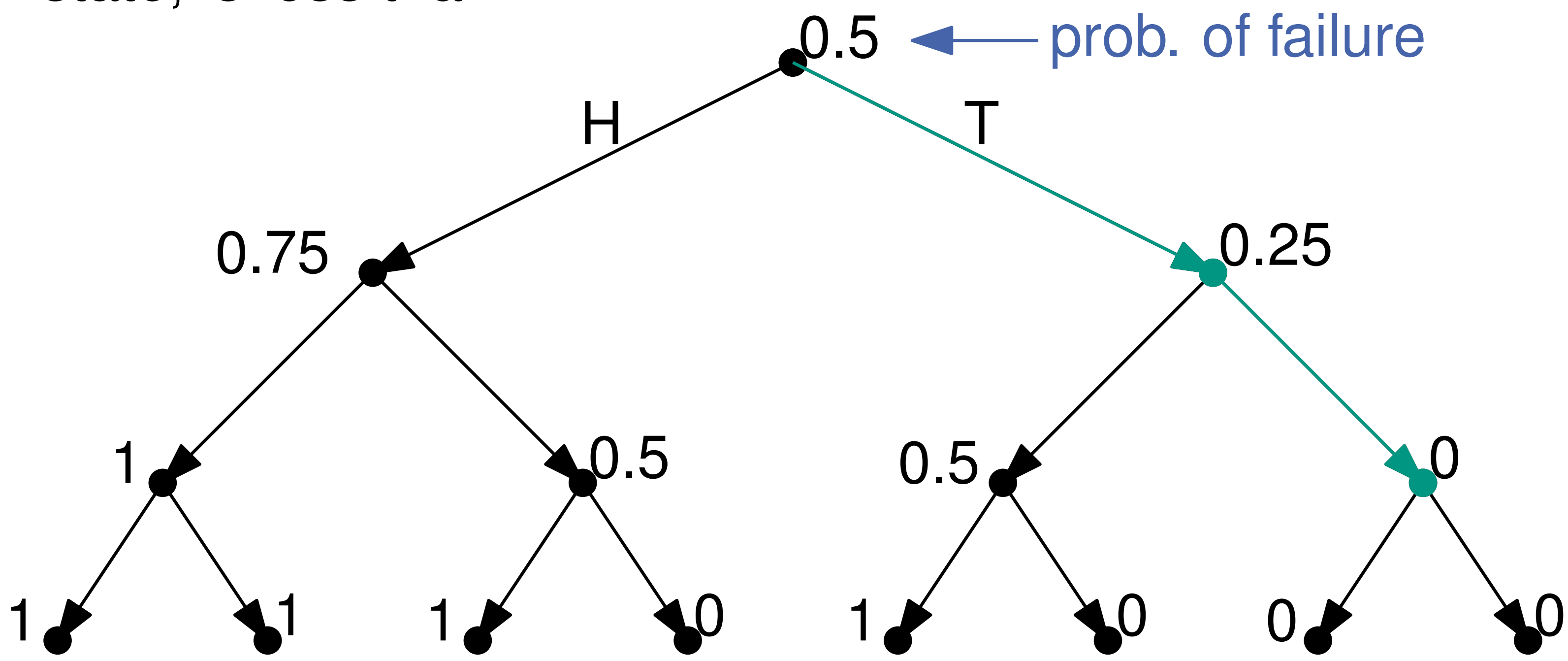
Method of Conditional Probabilities

- Convert proof via the probabilistic method to efficient, deterministic alogrithm
- replace random root-to-leaf walk with deterministic walk
- maintain invariant: the conditional probability of failure, given the current state, is less than 1



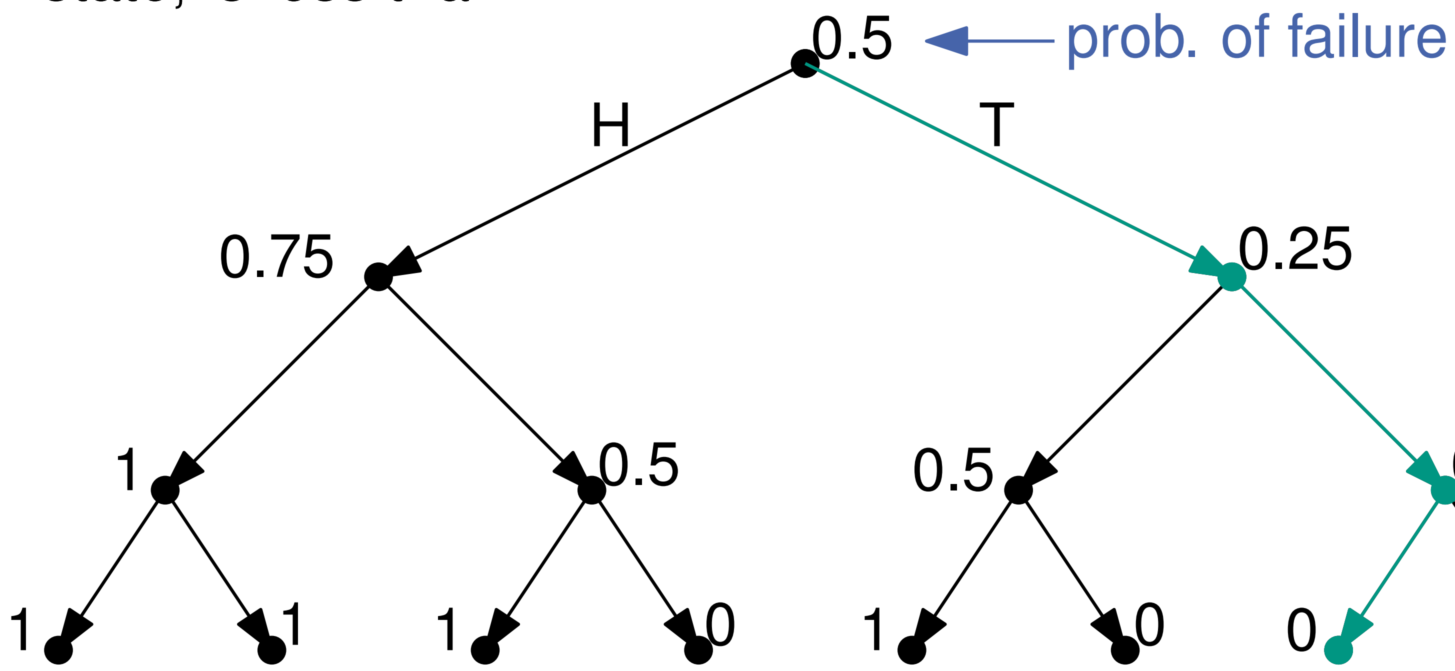
Method of Conditional Probabilities

- Convert proof via the probabilistic method to efficient, deterministic alogrithm
- replace random root-to-leaf walk with deterministic walk
- maintain invariant: the conditional probability of failure, given the current state, is less than 1



Method of Conditional Probabilities

- Convert proof via the probabilistic method to efficient, deterministic alogrithm
- replace random root-to-leaf walk with deterministic walk
- maintain invariant: the conditional probability of failure, given the current state, is less than 1



Institute of Theoretical Informatics
Algorithmics Group

Deterministic Algorithm: Overview

Algorithm 1: deterministic algorithm

initially no pillars are built;
initialize objective function value;
for *offset* $k = 1, \ldots, \ell$ **do**
    calculate Δ_0 for not building pillar at k;
    calculate Δ_1 for not building pillar at k;
    choose option minimizing obj. function;

Deterministic Algorithm

Karlsruhe Institute of Technology

- let $X_1, \dots, X_\ell$ be zero-one random variables with $\mathbb{P}[X_i = 1] = \frac{\ln n}{n}$
- let $x_1, \dots, x_\ell$ be given values for the random variables

Deterministic Algorithm

- let $X_1, \ldots, X_\ell$ be zero-one random variables with $\mathbb{P}[X_i = 1] = \frac{\ln n}{n}$
- let $x_1, \ldots, x_\ell$ be given values for the random variables
- $T_k = \{i | x_i = 1\} \cup \{j | X_j = 1\}$
- p_i = probability that train falls through track at position i

Institute of Theoretical Informatics
Algorithmics Group

Deterministic Algorithm

- let $X_1, \ldots, X_\ell$ be zero-one random variables with $\mathbb{P}[X_i = 1] = \frac{\ln n}{n}$
- let $x_1, \ldots, x_\ell$ be given values for the random variables
- $T_k = \{i | x_i = 1\} \cup \{j | X_j = 1\}$
- p_i = probability that train falls through track at position i
- T adds a pillar for each position j where the train falls through the track

Institute of Theoretical Informatics
Algorithmics Group

Deterministic Algorithm

- let $X_1, \ldots, X_\ell$ be zero-one random variables with $\mathbb{P}[X_i = 1] = \frac{\ln n}{n}$
- let $x_1, \ldots, x_\ell$ be given values for the random variables
- $T_k = \{i | x_i = 1\} \cup \{j | X_j = 1\}$
- p_i = probability that train falls through track at position i
- T adds a pillar for each position j where the train falls through the track
- let $F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_\ell) = |T|$ be the objective function

Institute of Theoretical Informatics
Algorithmics Group

Deterministic Algorithm

- $\mathbb{E}[F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_\ell)] = \{i | x_i = 1\} + \frac{\ln n}{n}(\ell - k) + \sum_i p_i$

Already built pillars

exp. additional pillars in T_k

$\mathbb{E}[|T_k|]$

exp. #positions where train falls through

$\mathbb{E}[T \setminus T_k]$

Deterministic Algorithm

- $\mathbb{E}[F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_\ell)] = \{i | x_i = 1\} + \frac{\ln n}{n}(\ell - k) + \sum_i p_i$

  Already built pillars

  exp. additional pillars in T_k

  exp. #positions where train falls through

  $\mathbb{E}[|T_k|]$

  $\mathbb{E}[T \setminus T_k]$

- choosing 1: $\mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] + 1 - \frac{\ln n}{n}$, $p_i = 0$ where $k + 1 \in (C + i)$

Deterministic Algorithm

- $\mathbb{E}[F(x_1, \dots, x_k, X_{k+1}, \dots, X_\ell)] = \{i | x_i = 1\} + \frac{\ln n}{n}(\ell - k) + \sum_i p_i$

Already built pillars

exp. additional pillars in T_k

exp. #positions where train falls through

$$\underbrace{\text{Already built pillars} + \text{exp. additional pillars in } T_k}_{\mathbb{E}[|T_k|]} \qquad \underbrace{\text{exp. \#positions where train falls through}}_{\mathbb{E}[T \setminus T_k]}$$

- choosing 1: $\mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] + 1 - \frac{\ln n}{n}$, $p_i = 0$ where $k + 1 \in (C + i)$
- choosing 0: $\mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] - \frac{\ln n}{n}$, update affected p_i with $\frac{p_i}{1 - (\ln n)/n}$

Institute of Theoretical Informatics
Algorithmics Group

Deterministic Algorithm

Karlsruhe Institute of Technology

- $\mathbb{E}[F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_\ell)] = \{i | x_i = 1\} + \frac{\ln n}{n}(\ell - k) + \sum_i p_i$

Already built pillars

exp. additional pillars in T_k

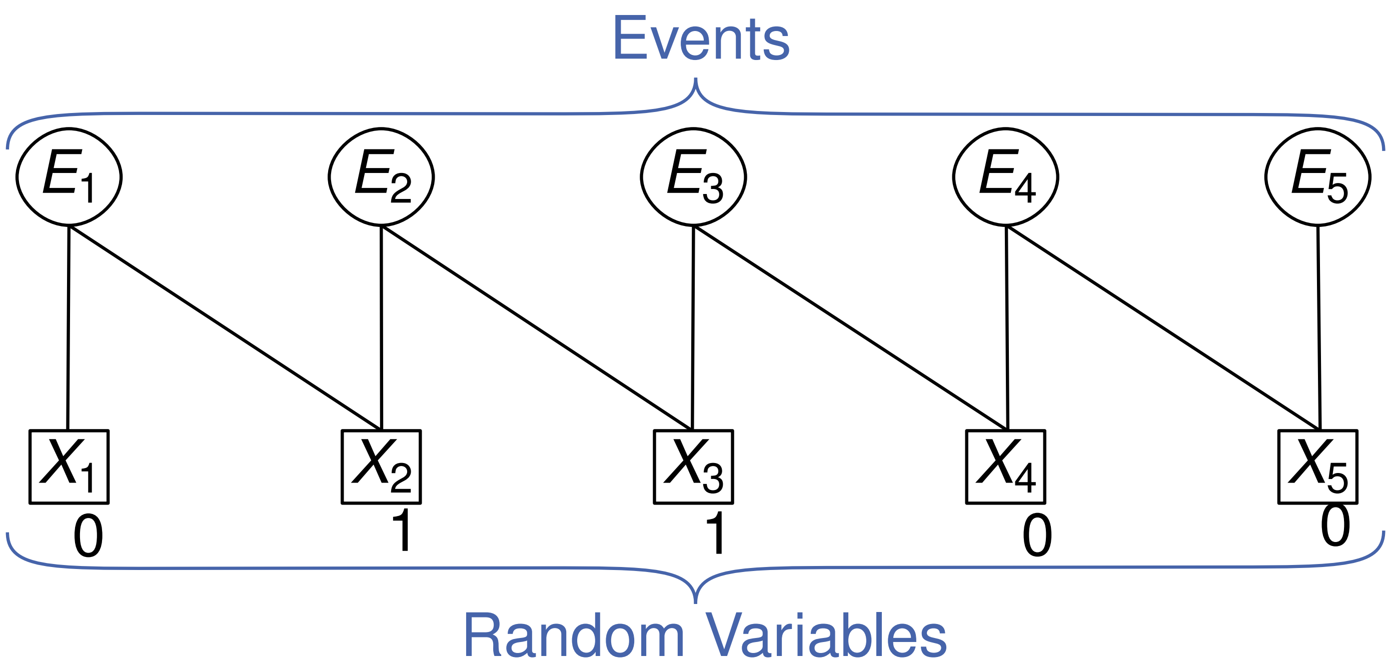
exp. #positions where train falls through

$\mathbb{E}[|T_k|]$

$\mathbb{E}[T \setminus T_k]$

- choosing 1: $\mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] + 1 - \frac{\ln n}{n}$, $p_i = 0$ where $k + 1 \in (C + i)$
- choosing 0: $\mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] - \frac{\ln n}{n}$, update affected p_i with $\frac{p_i}{1-(\ln n)/n}$
- correctness:
- runtime: ℓ iterations with $\mathcal{O}(n)$
- track length: arbitrary wheel arrangement proof showed $\mathbb{E}[F(x_1, \ldots, x_k, X_{k+1}, \ldots, X_\ell)] \leq \frac{1+\ln n}{n}$

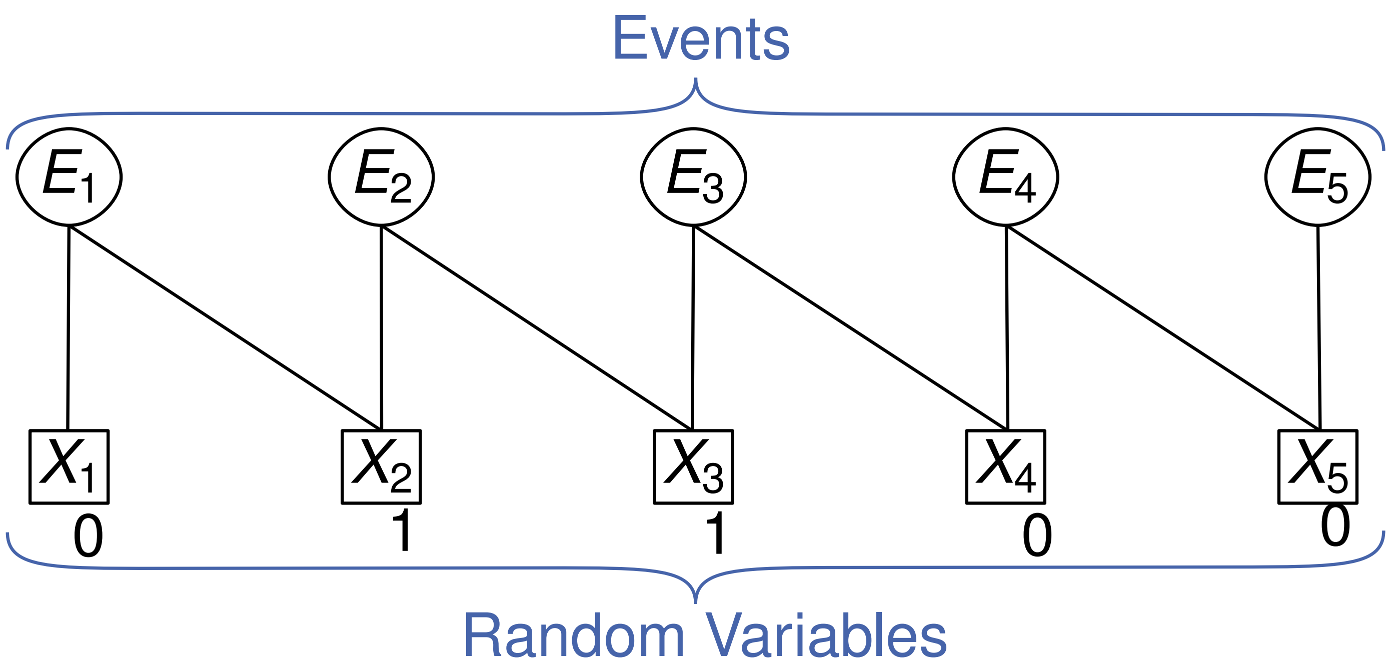
Lovász Local Lemma (LLL)



- connected X_i determine event outcome
- d = #events connected via path of length 2
- each event depends on at most d other events

Institute of Theoretical Informatics
Algorithmics Group

Lovász Local Lemma (LLL)



- connected X_i determine event outcome
- d = #events connected via path of length 2
- each event depends on at most d other events

Lemma: Given p with $\mathbb{P}[E_i] \leq p$ and $pde \leq 1$,
then $\mathbb{P}$[none of the events E_i occur] > 0

Algorithmic Lovász Local Lemma

Karlsruhe Institute of Technology

- given the LLL holds
- the **fix-it algorithm** resamples an event E_i at most exp. $\frac{1}{d}$ times

---

Algorithm 2: fix-it algorithm

Data: independent random variables $X_1, \ldots, X_s$, events $E_1, \ldots, E_m$;

```
Independently sample each X_1, ..., X_s;
while ∃E_i that holds do
    select E_i;
    resample all X_j, E_i depends on;
```

$\Rightarrow$ exp. $\frac{m}{d}$ iterations

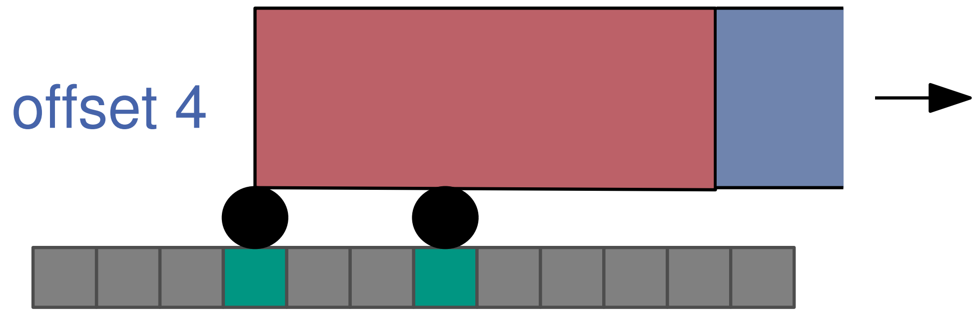
Institute of Theoretical Informatics
Algorithmics Group

Fix-it Algorithm for Train Tracks

- $\mathbb{P}[X_i = 1] = \frac{1+2\ln n}{n}$ decides wether to build pillar i
- E_i = event that train falls through tracks at offset i
- what is d?

Fix-it Algorithm for Train Tracks

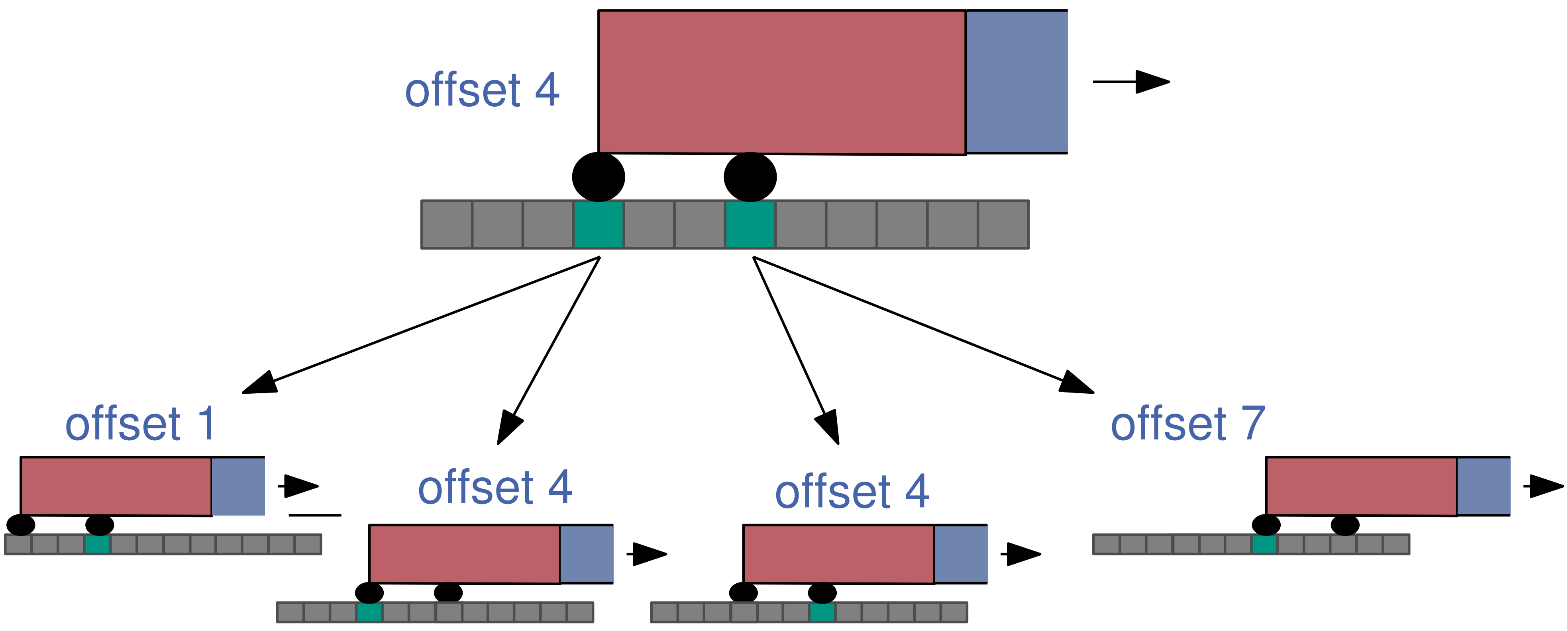
- $\mathbb{P}[X_i = 1] = \frac{1+2\ln n}{n}$ decides wether to build pillar i
- E_i = event that train falls through tracks at offset i
- what is d?



Institute of Theoretical Informatics
Algorithmics Group

Fix-it Algorithm for Train Tracks

- $\mathbb{P}[X_i = 1] = \frac{1+2\ln n}{n}$ decides wether to build pillar i
- E_i = event that train falls through tracks at offset i
- what is d?



Institute of Theoretical Informatics
Algorithmics Group

Fix-it Algorithm for Train Tracks

- $\mathbb{P}[X_i = 1] = \frac{1+2\ln n}{n}$ decides wether to build pillar i
- E_i = event that train falls through tracks at offset i
- $d = n^2$
- $\mathbb{P}[E_i] = (1 - \frac{1+2\ln n}{n})^n < \frac{1}{e^{1+2\ln n}} \leq \frac{1}{en^2} = p$
- $pde = \frac{1}{en^2} n^2 e = 1$

Institute of Theoretical Informatics
Algorithmics Group

Fix-it Algorithm for Train Tracks

- $\mathbb{P}[X_i = 1] = \frac{1+2\ln n}{n}$ decides wether to build pillar i
- E_i = event that train falls through tracks at offset i
- $d = n^2$
- $\mathbb{P}[E_i] = (1 - \frac{1+2\ln n}{n})^n < \frac{1}{e^{1+2\ln n}} \leq \frac{1}{en^2} = p$
- $pde = \frac{1}{en^2} n^2 e = 1$
- correctness: ✔

Fix-it Algorithm for Train Tracks

- $\mathbb{P}[X_i = 1] = \frac{1+2\ln n}{n}$ decides wether to build pillar i
- E_i = event that train falls through tracks at offset i
- $d = n^2$
- $\mathbb{P}[E_i] = (1 - \frac{1+2\ln n}{n})^n < \frac{1}{e^{1+2\ln n}} \leq \frac{1}{en^2} = p$
- $pde = \frac{1}{en^2} n^2 e = 1$
- correctness: ✔
- runtime: Algorithmic LLL $\Rightarrow \frac{\ell}{n^2}$ iterations each naively $\mathcal{O}(n^3)$ ✔

Fix-it Algorithm for Train Tracks

- $\mathbb{P}[X_i = 1] = \frac{1+2\ln n}{n}$ decides wether to build pillar i
- E_i = event that train falls through tracks at offset i
- $d = n^2$
- $\mathbb{P}[E_i] = (1 - \frac{1+2\ln n}{n})^n < \frac{1}{e^{1+2\ln n}} \leq \frac{1}{en^2} = p$
- $pde = \frac{1}{en^2} n^2 e = 1$
- correctness: ✔
- runtime: Algorithmic LLL $\Rightarrow \frac{\ell}{n^2}$ iterations each naively $\mathcal{O}(n^3)$ ✔
- track length:
  - worst case each resample gives n ones $\Rightarrow \mathcal{O}(\frac{\ell}{n})$
  - initial: $\mathcal{O}(\frac{\ell+2\ell\ln n}{n})$ ✔

Institute of Theoretical Informatics
Algorithmics Group

Min-Hash

- given collection of sets $\mathcal{S}$
- technique for sampling one element for each set $S \in \mathcal{S}$
- hash elements $h(s) \in (0, 1)$
- select element with minimum hash

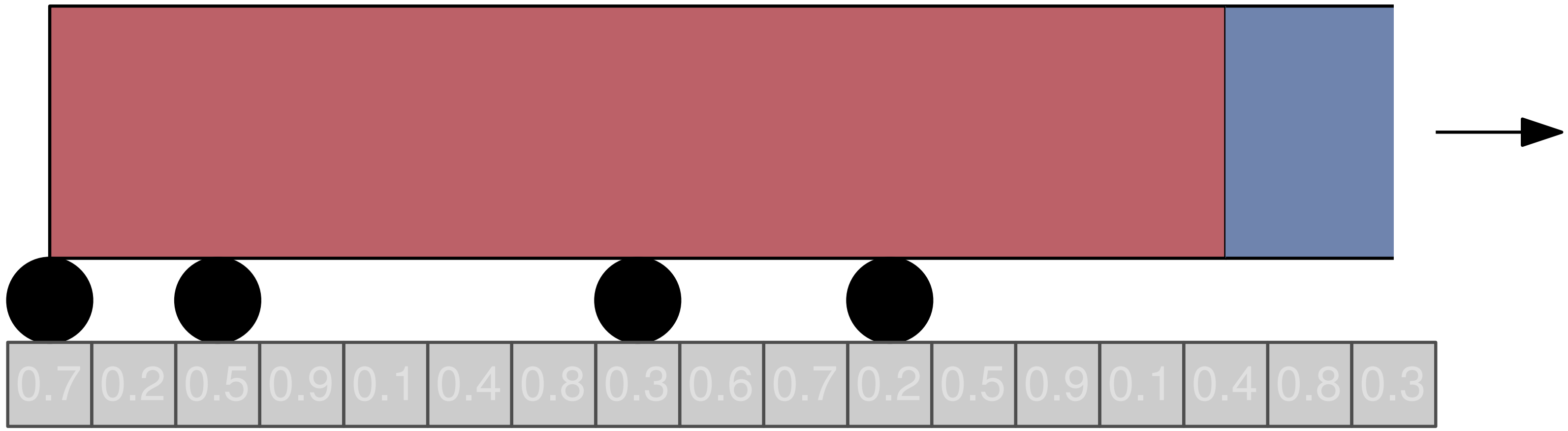
Institute of Theoretical Informatics
Algorithmics Group

Min-Hash

- given collection of sets $\mathcal{S}$
- technique for sampling one element for each set $S \in \mathcal{S}$
- hash elements $h(s) \in (0, 1)$
- select element with minimum hash

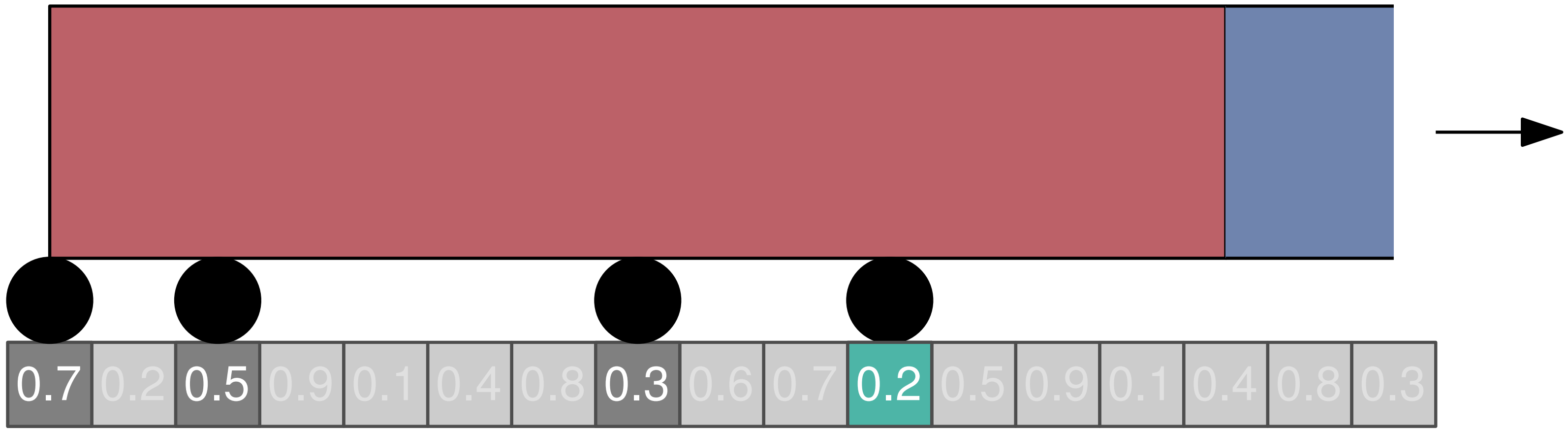
- Two important Properties
  - if sets S_1 and S_2 are similar, then their min-hash is likely to be the same
  - if $s \in S$ is the minimum-hashed element in one set it is likely to be the minimum-hashed element in other sets

The Algorithm



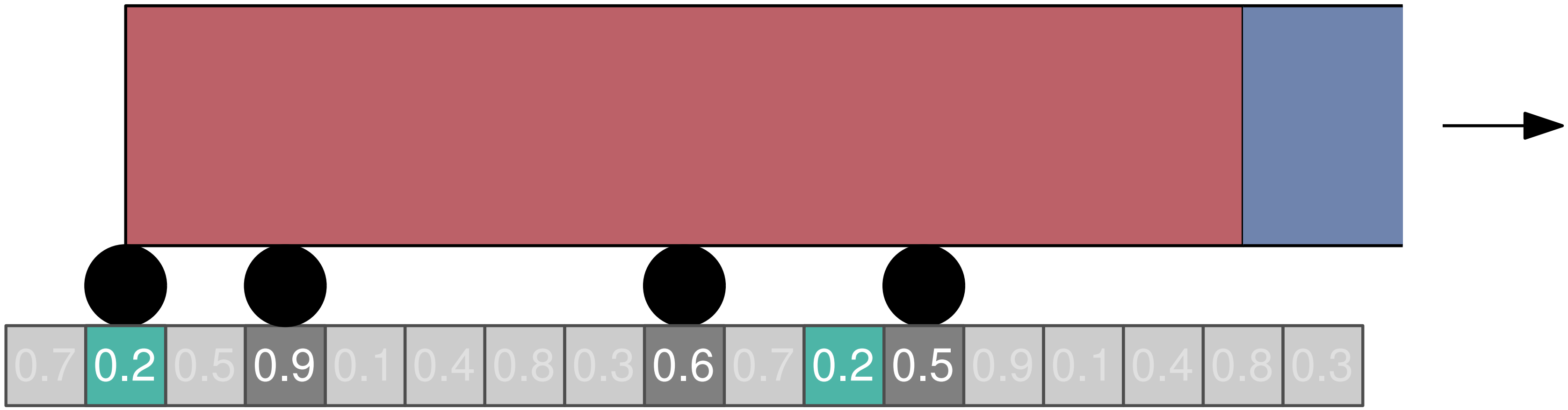
- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position

The Algorithm



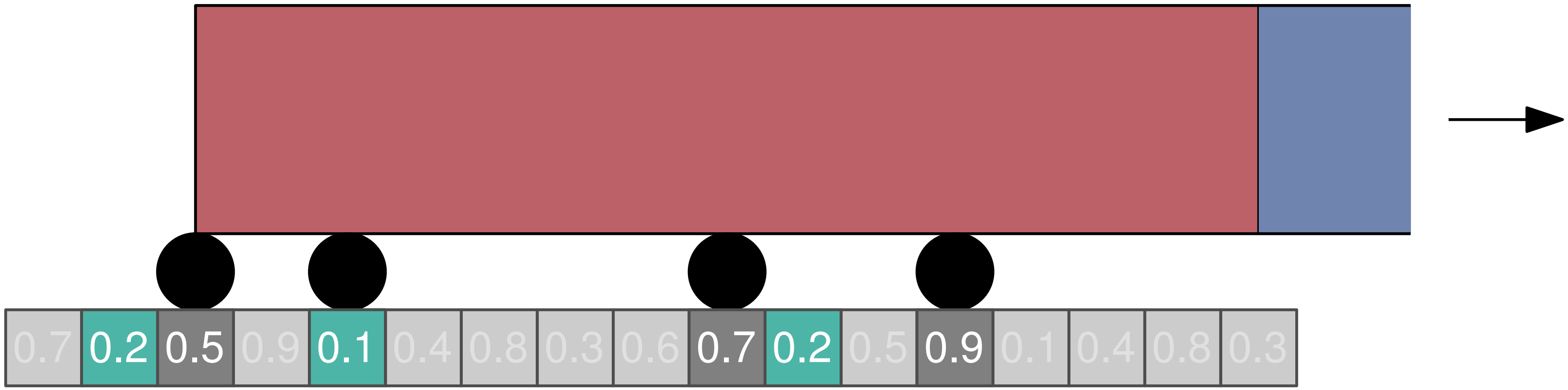
- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position

The Algorithm



- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position

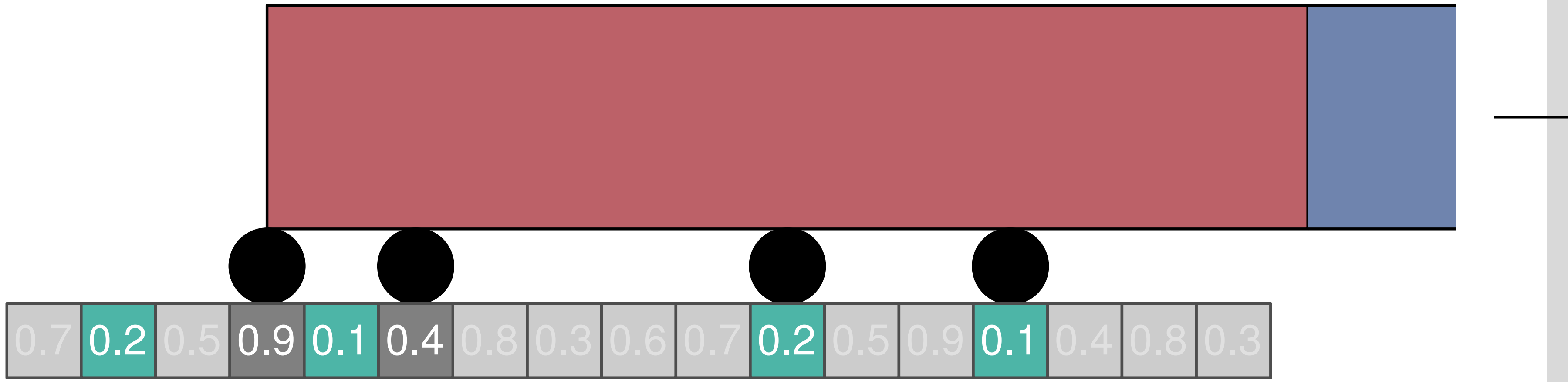
The Algorithm



- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position

Institute of Theoretical Informatics
Algorithmics Group

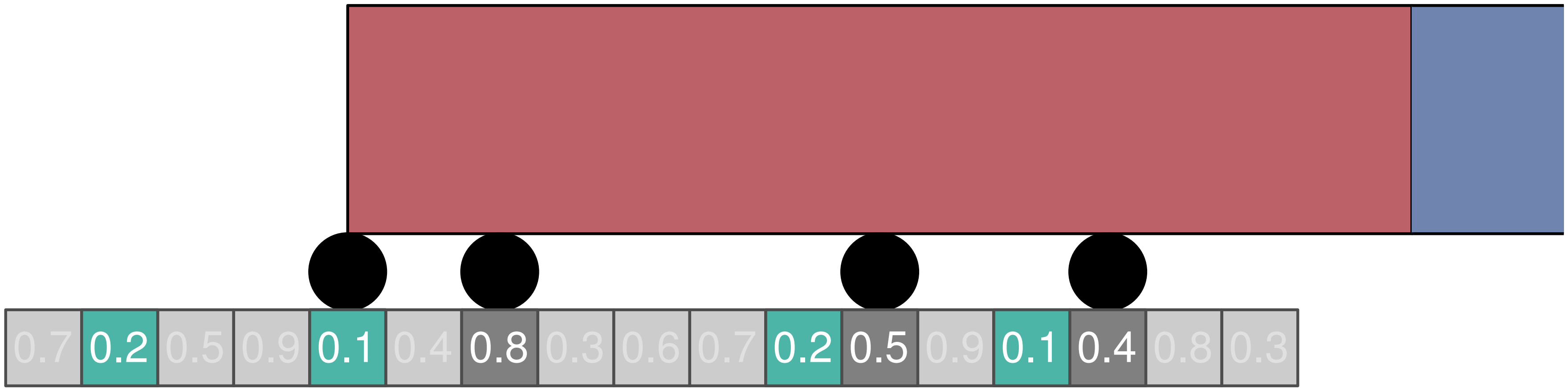
The Algorithm



- assign random real numbers $r_1, \ldots, r_\ell \in (0,1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position

Institute of Theoretical Informatics
Algorithmics Group

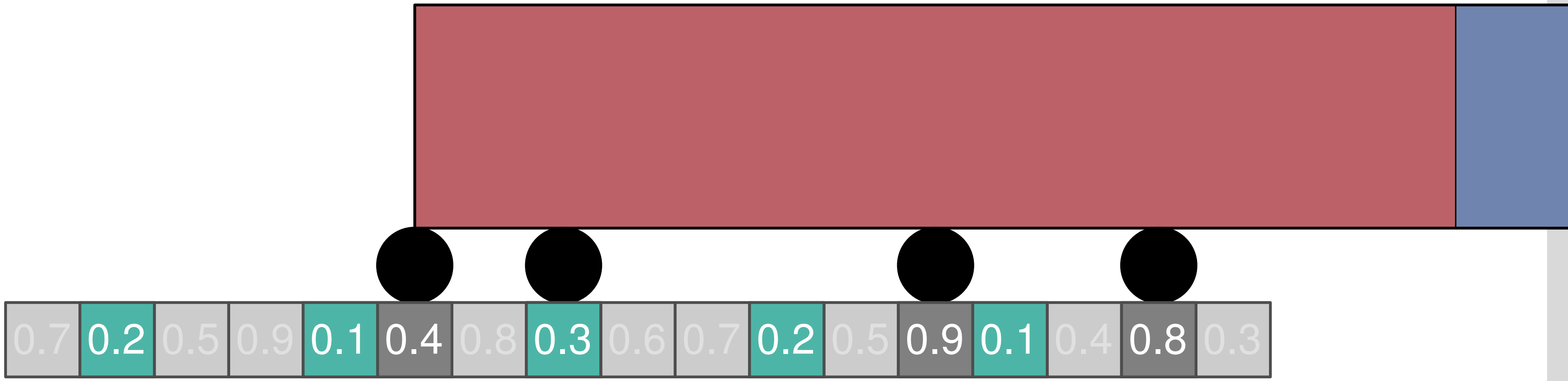
The Algorithm



- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position

Institute of Theoretical Informatics
Algorithmics Group

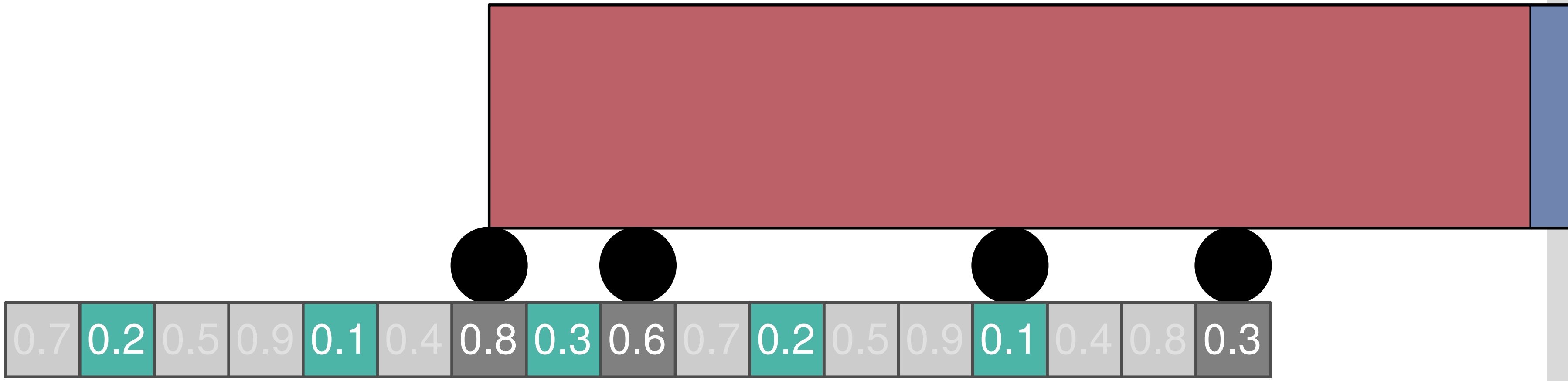
The Algorithm



- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position

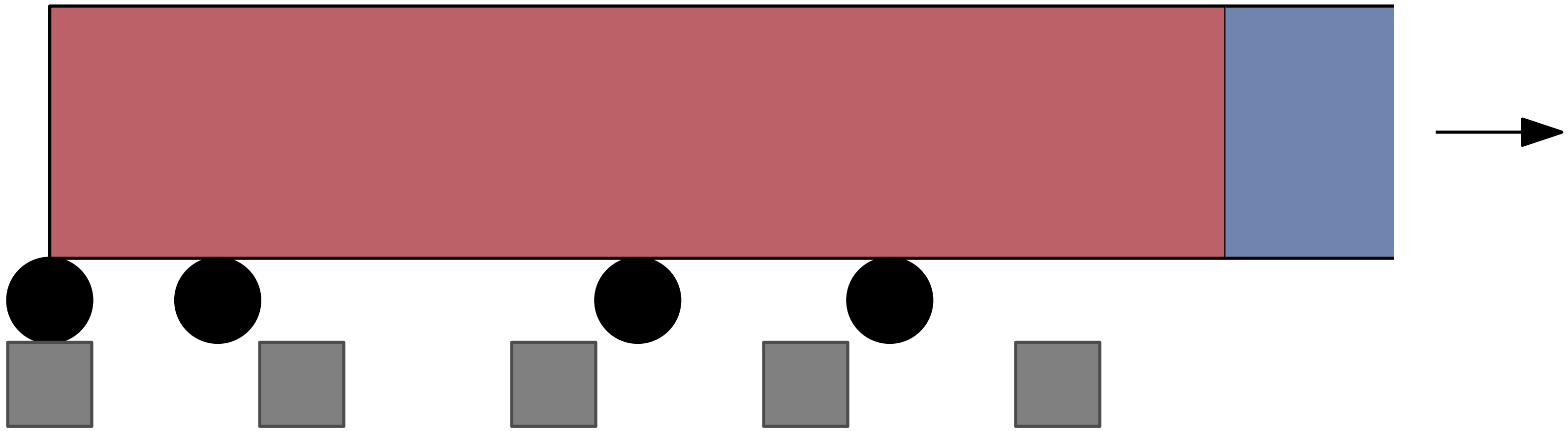
Institute of Theoretical Informatics
Algorithmics Group

The Algorithm



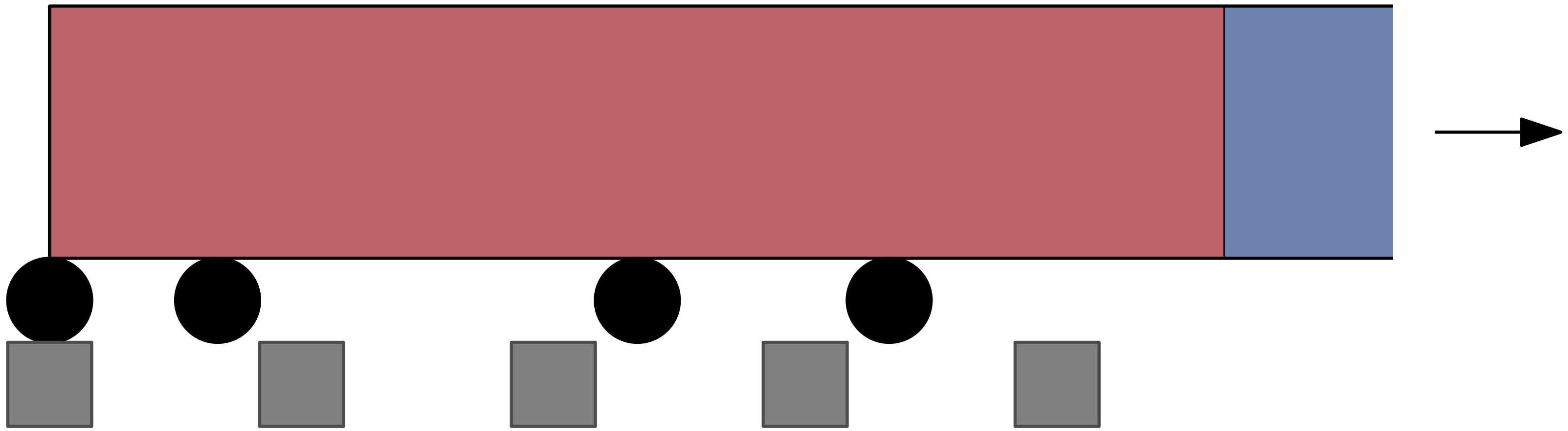
- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position

The Algorithm



- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position
- correctness: ✓
- runtime $\mathcal{O}(\ell n)$: ✓

The Algorithm



- assign random real numbers $r_1, \ldots, r_\ell \in (0, 1)$ to each possible track pillar
- iterate over all possible positions
- choose the pillar with the smallest hash that is at a wheel position
- correctness: ✓
- runtime $\mathcal{O}(\ell n)$: ✓
- track length: ?

Proof: Min-Hashing Algorithm

- $S_k = (C + k)$ the wheel positions of the rear quarter at offset k

Institute of Theoretical Informatics
Algorithmics Group

Proof: Min-Hashing Algorithm

- $S_k = (C + k)$ the wheel positions of the rear quarter at offset k
- selected pillar: $\arg\min_{s \in S_k} r_s$
- key observation: almost all sampled pillars have small r_s

Institute of Theoretical Informatics
Algorithmics Group

Proof: Min-Hashing Algorithm

- $S_k = (C + k)$ the wheel positions of the rear quarter at offset k
- selected pillar: $\arg\min_{s \in S_k} r_s$
- key observation: almost all sampled pillars have small r_s
- let $r_s > \frac{\ln n}{n}$
  $\Rightarrow$ all other $s' \in S$ have larger values
- prob. of selecting such a pillar: $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- the expected number of pillars then is $\frac{\ell - f}{n} \leq \frac{\ell}{n}$

Proof: Min-Hashing Algorithm

- $S_k = (C + k)$ the wheel positions of the rear quarter at offset k
- selected pillar: $\arg\min_{s \in S_k} r_s$
- key observation: almost all sampled pillars have small r_s
- let $r_s > \frac{\ln n}{n}$
  $\Rightarrow$ all other $s' \in S$ have larger values
- prob. of selecting such a pillar: $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- the expected number of pillars then is $\frac{\ell - f}{n} \leq \frac{\ell}{n}$
- exp number of pillars with $r_s \leq \frac{\ln n}{n}$ is at most $\frac{\ell \ln n}{n}$

Proof: Min-Hashing Algorithm

- $S_k = (C + k)$ the wheel positions of the rear quarter at offset k
- selected pillar: $\arg\min_{s \in S_k} r_s$
- key observation: almost all sampled pillars have small r_s
- let $r_s > \frac{\ln n}{n}$
  $\Rightarrow$ all other $s' \in S$ have larger values
- prob. of selecting such a pillar: $(1 - \frac{\ln n}{n})^n \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$
- the expected number of pillars then is $\frac{\ell - f}{n} \leq \frac{\ell}{n}$
- exp number of pillars with $r_s \leq \frac{\ln n}{n}$ is at most $\frac{\ell \ln n}{n}$
  $\Rightarrow$ overall $\leq \frac{\ell(1+\ln n)}{n}$

What have we learned?

- Probabilistic Method and its applications for train tracks

Institute of Theoretical Informatics
Algorithmics Group

What have we learned?

- Probabilistic Method and its applications for train tracks
- $\mathcal{O}(\ell/n)$ track for equally spaced wheels

Institute of Theoretical Informatics
Algorithmics Group

What have we learned?

- Probabilistic Method and its applications for train tracks
- $\mathcal{O}(\ell/n)$ track for equally spaced wheels
- $\mathcal{O}(\frac{\ell \ln n}{n})$ track for arbitrary wheel arrangements

Institute of Theoretical Informatics
Algorithmics Group

What have we learned?

- Probabilistic Method and its applications for train tracks
- $\mathcal{O}(\ell/n)$ track for equally spaced wheels
- $\mathcal{O}(\frac{\ell \ln n}{n})$ track for arbitrary wheel arrangements
- Derandomisation using the method of conditional probabilities

What have we learned?

- Probabilistic Method and its applications for train tracks
- $\mathcal{O}(\ell/n)$ track for equally spaced wheels
- $\mathcal{O}(\frac{\ell \ln n}{n})$ track for arbitrary wheel arrangements
- Derandomisation using the method of conditional probabilities
- Fix-it algorithm based on algorithmic Lovász Local Lemma

Institute of Theoretical Informatics
Algorithmics Group

What have we learned?

- Probabilistic Method and its applications for train tracks
- $\mathcal{O}(\ell/n)$ track for equally spaced wheels
- $\mathcal{O}(\frac{\ell \ln n}{n})$ track for arbitrary wheel arrangements
- Derandomisation using the method of conditional probabilities
- Fix-it algorithm based on algorithmic Lovász Local Lemma
- Min-Hash based algorithm

Appendix

Deterministic Algorithm: Correctness

- why is the result not $F(0, \ldots, 0) \leq \frac{1+\ln n}{n}$?
- let X be the set of pillars the event E_i depends on
- let $0 = x_i \in X, i \neq k$
- $\Delta_0 = -\frac{\ln n}{n} + \Delta_{\sum p_i} \geq -\frac{\ln n}{n} + (1 - \frac{\ln n}{n}) > 0$
- $\Delta_1 = 1 - \frac{\ln n}{n} - (1 - \frac{\ln n}{n}) = 0$

- choosing 1: $\mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] + 1 - \frac{\ln n}{n}$ and zeroing out all p_i where $x_k \in (C + i)$
- choosing 0: $\mathbb{E}[|T_{k+1}|] = \mathbb{E}[|T_k|] - \frac{\ln n}{n}$ and updating affected p_i with $\frac{p_i}{1-(\ln n)/n}$