Rainbow Sort

– Sorting at the Speed of Light –

Dominik Schultes

31. May 2004

1. Introduction: Complexity of Sorting

2. Rainbow Sort – Idea

3. Rainbow Sort – Implementation

Upper Bounds for Sorting

	Input	Processing	Output	Σ
Heapsort	n	$n \log n$	n	$\Theta(n \log n)$
Counting Sort	n	$n + m$	n	$\Theta(n + m)$
Bead Sort	n	$\sqrt{n}$	n	$\Theta(n)$

Note: Space Complexity of Bead Sort = $\Theta(n \cdot m)$

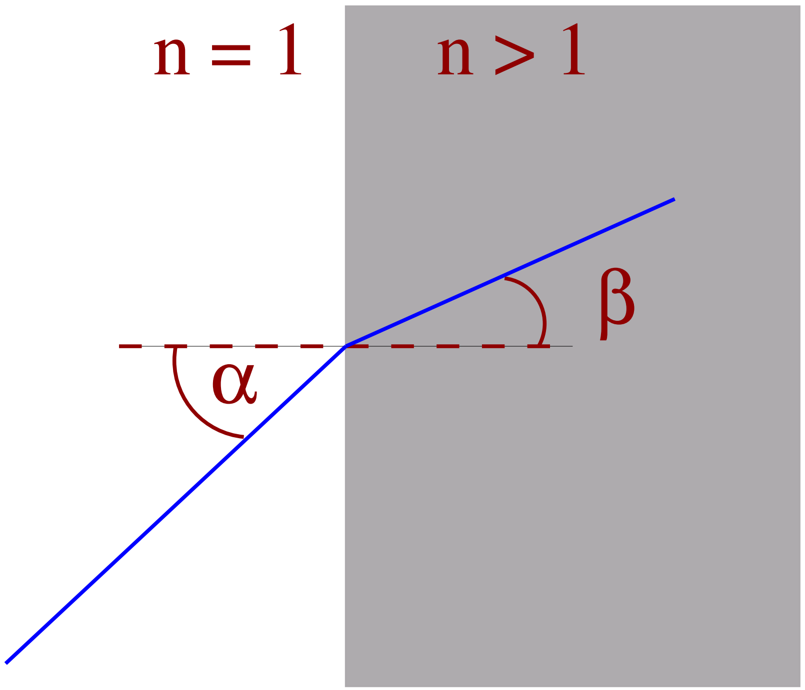
n = # items
m = maximum key value

Lower Bounds for Sorting

- comparison-based sorting: $\Omega(n \log n)$

- in general, sorting: $\Omega(n)$

Basics: Refraction

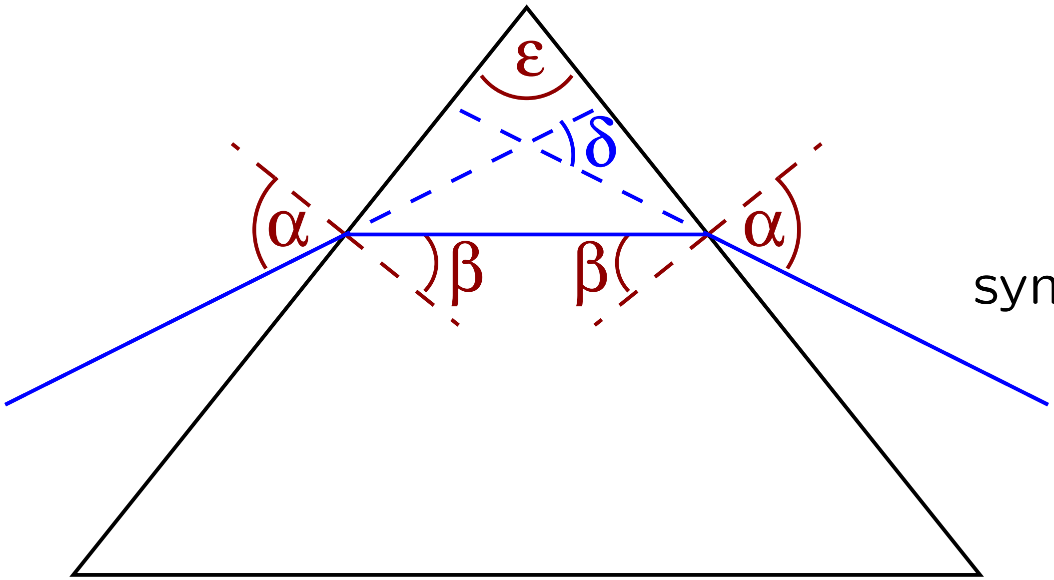


Snell's Law

$$\frac{\sin\alpha}{\sin\beta} = \frac{c}{c_n} = n$$

in this context: n = refraction index

Basics: Prism



symmetrical optical path

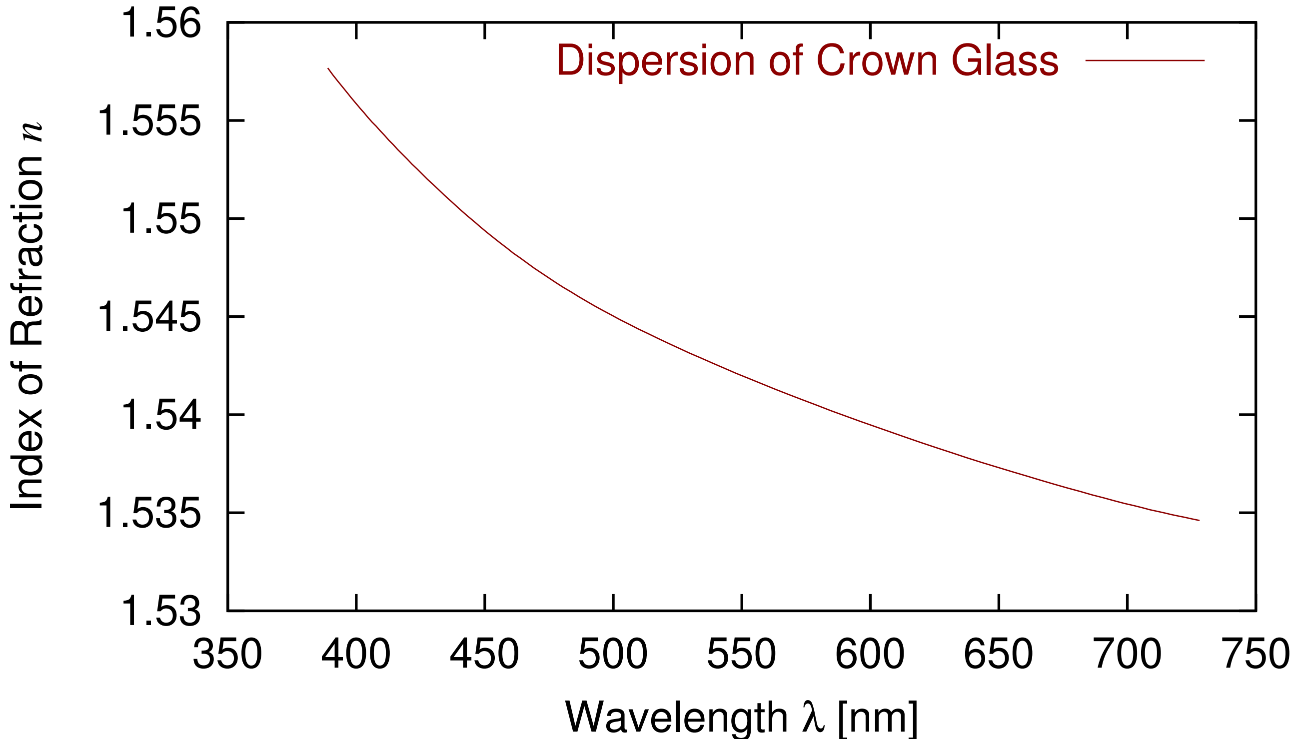
$$\text{angular deviation} : \delta = 2 \arcsin\left(n \sin\frac{\varepsilon}{2}\right) - \varepsilon$$

arcsin is strictly monotonic increasing

$\rightsquigarrow$ the greater n, the greater the deviation

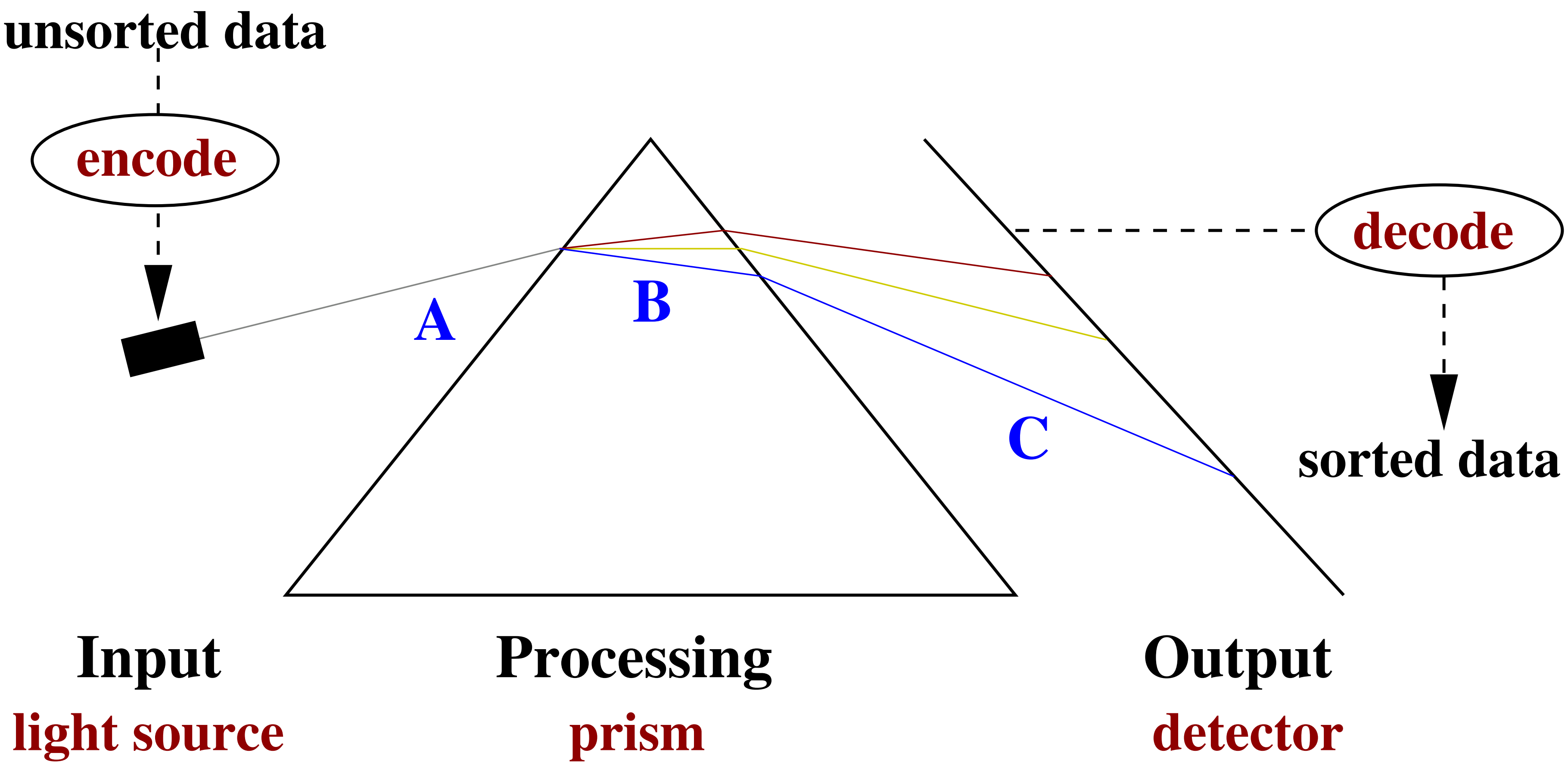
Basics: Dispersion

Refraction index depends on the wavelength of the ray



the less the wavelength λ, the greater n, the greater the deviation

Setup


unsorted data
encode
decode
A
B
C
sorted data
Input
light source
Processing
prism
Output
detector

Encoding / Decoding

Encoding: use s. m. increasing function $f : [0, m+1] \rightarrow [\lambda_{min}, \lambda_{max}]$

Decoding: use f^{-1}

Encoding of Duplicates: use $f(x + i/n)$ instead of $f(x)$
where $x =$ the number, $i =$ position in the input

for the sake of simplicity:

Assumption: no duplicates

Algorithm

Input

`Ray` $ray := \emptyset$

for each $x \in Input$ **do** $ray := ray \cup f(x)$

Processing

send ray through prism

Output

`Stack` $sorted := \emptyset$

`Wavelength` $cur\lambda := \infty$

whenever $\min \lambda(incoming\ rays) < cur\lambda$ **do**

  $cur\lambda := \min \lambda(incoming\ rays)$

  $sorted.\mathrm{push}(f^{-1}(cur\lambda))$

  if $sorted.\mathrm{size} = n$ **then return** $sorted$

Simulator

Correctness

Let be $\lambda_{min} \leq \lambda_1 < \lambda_2 \leq \lambda_{max}$. Then λ_2 arrives before λ_1.

Proof:

$$\lambda_1 < \lambda_2$$

$$\rightsquigarrow n(\lambda_1) > n(\lambda_2)$$

$$\rightsquigarrow \quad \delta(\lambda_1) > \delta(\lambda_2) \quad (\rightsquigarrow \text{ longer path for } \lambda_1)$$

$$\wedge \quad c(\lambda_1) < c(\lambda_2) \quad (\rightsquigarrow \lambda_1 \text{ slower in the prism})$$

path / speed	λ_1	λ_2
A	equal / equal	equal / equal
B	longer / slower	shorter / faster
C	longer / equal	shorter / equal

Correctness (cont'd)

$\rightsquigarrow$ Wavelengths arrive in decreasing order.

$\rightsquigarrow$ Whenever a new wavelength arrives, it is smaller than $cur\lambda$.

$\rightsquigarrow$ $cur\lambda$ equals to all wavelengths one after the other in decr. order.

$\rightsquigarrow$ Output is complete (because coding function f is bijective)
$\wedge$ Output is sorted (because f^{-1} is s. m. increasing)

Complexity

Input	$\Omega(n)$	$O(?)$
Processing	$\Theta(1)$	
Output	$\Omega(n)$	$O(?)$
Space	$\Omega(n)$	$O(?)$

Heisenberg uncertainty principle

$$\Delta W \cdot \Delta t \geq \frac{h}{2\pi}$$

the more precise the measurement of the energy W ($\sim 1/\lambda$), the more time t is needed

Input: Laser

Use laser that can be tuned continuously over a range of wavelengths $[\lambda_{min}, \lambda_{max}]$.

Difficulty: precise setting

- if $O(1)$ is sufficient, input $\in \Theta(n)$
- if we need to measure, input $\in \Theta(n + m)$

Output: Detector

most difficult part !

One possibility:
determine wavelength by measuring energy
$\leadsto$ output $\in \Theta(n + m^2)$

Proof:
T = running time, ℓ = length of the path of λ_{min},
d = distance between λ_{min} and λ_{max} at the detector,
Δd = distance between two adjacent wavelengths at the detector

(1) $T \sim \ell \sim d$ (2) $d = \Delta d \cdot m$ (3) $\Delta d \sim \Delta t \sim 1/\Delta W \sim m$

$$\leadsto T \sim \Delta d \cdot m \sim m \cdot m$$

Output: Detector (cont'd)

Another possibility:

determine wavelength by the point of contact of the incoming ray (the detector is subdivided into m cells)

$\leadsto$ measurement of the energy not required

$\leadsto$ output $\in \Theta(n+m)$

Proof:

(1) $T \sim \ell \sim d$ (2) $d = \Delta d \cdot m$ (3) $\Delta d = \text{const}$

$$\leadsto T \sim m$$

Note: Space $\in \Theta(n+m)$

Conclusion

Input	$\Omega(n)$	$O(n+m)$
Processing	$\Theta(1)$	
Output	$\Omega(n)$	$O(n+m)$
Space	$\Omega(n)$	$O(n+m)$

- if measurement required: $\Theta(n+m)$
- otherwise: somewhere between $\Omega(n)$ and $O(n+m)$

in general: lower bound for sorting (time / space) ?