Beyond Simple Aggregates: Indexing for Summary Queries

Zhewei Wei and Ke Yi

Hong Kong University of Science and Technology

Reporting vs. Aggregation

```
SELECT salary
FROM Table T
WHERE 30 < age < 40
```

Reporting vs. Aggregation

```
SELECT salary
FROM Table T
WHERE 30 < age < 40
```

$$\left.\begin{array}{c}\$32,000\\ \$76,300\\ \$54,400\\ \ldots\\ \$68,000\\ \$28,000\end{array}\right\}\text{50,000 records}$$

Reporting vs. Aggregation

```
SELECT salary
FROM Table T
WHERE 30 < age < 40
```

$\$32,000$
$\$76,300$
$\$54,400$
$\ldots$
$\$68,000$
$\$28,000$

} 50,000 records

```
SELECT AVG(salary)
FROM Table T
WHERE 30 < age < 40
```

Reporting vs. Aggregation

```
SELECT salary
FROM Table T
WHERE 30 < age < 40
```

$$\left.\begin{array}{l}\$32,000\\ \$76,300\\ \$54,400\\ \ \ldots\\ \$68,000\\ \$28,000\end{array}\right\}\text{50, 000 records}$$

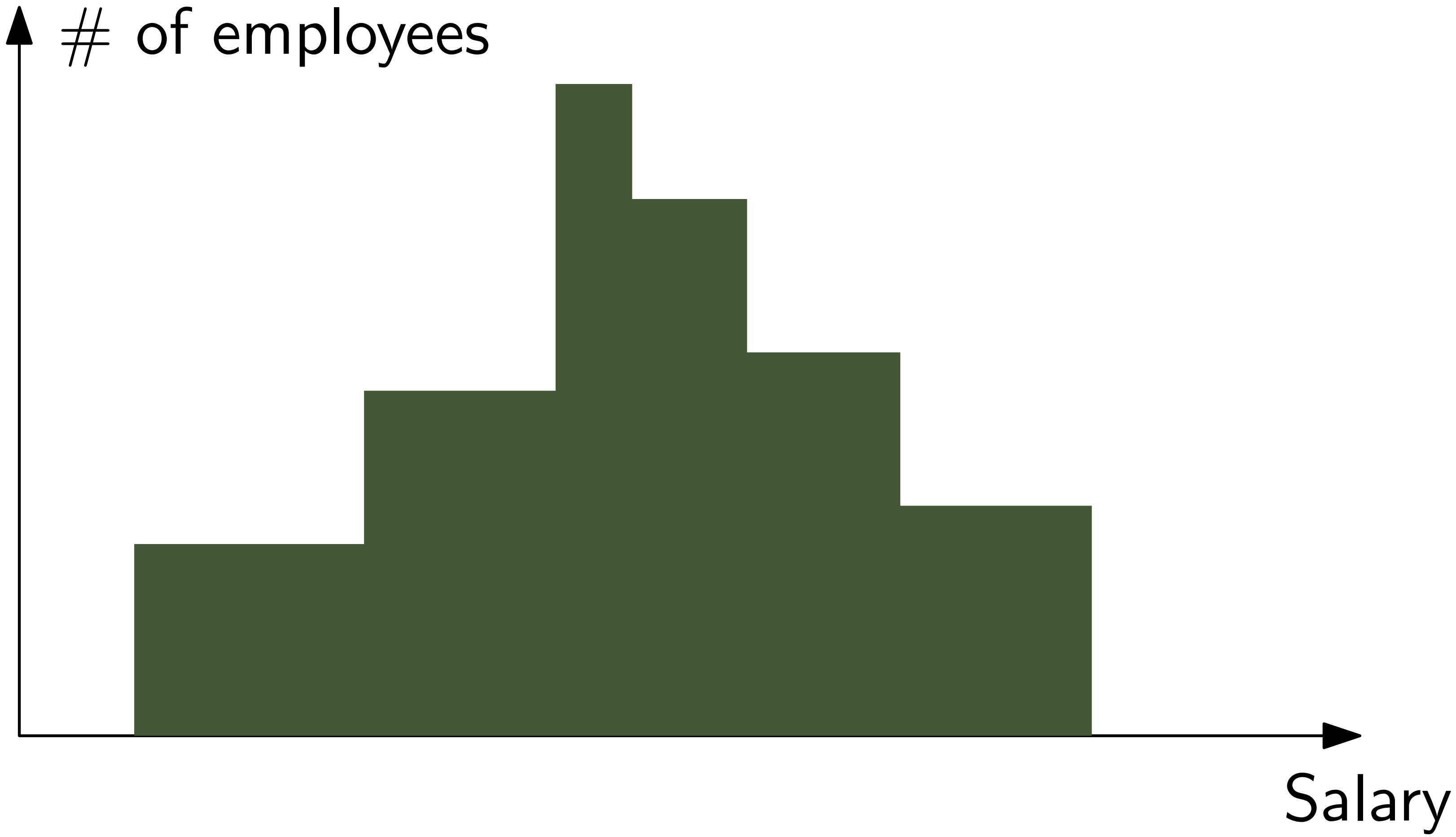
```
SELECT AVG(salary)
FROM Table T
WHERE 30 < age < 40
```

$52,312

Reporting vs. Aggregation

```
SELECT salary
FROM Table T
WHERE 30 < age < 40
```

```
SELECT AVG(salary)
FROM Table T
WHERE 30 < age < 40
```



Reporting vs. Aggregation

Search Engine Log

Date	Keyword
2011.04.08	Masters 2011
2011.04.08	Libya
2011.04.07	Japan nuclear crisis
2011.04.07	Libya
. . .	
2011.03.11	Japan earthquake
2011.03.11	Japan tsunami
2011.03.10	NCAA
. . .	

Reporting vs. Aggregation

Search Engine Log

Date	Keyword
2011.04.08	Masters 2011
2011.04.08	Libya
2011.04.07	Japan nuclear crisis
2011.04.07	Libya
. . .	
2011.03.11	Japan earthquake
2011.03.11	Japan tsunami
2011.03.10	NCAA
. . .	

$\Longrightarrow$

Keyword	Frequency
Libya	19.3%
Japan nuclear crisis	16.5%
Japan earthquake	10.2%
. . .	

Summary Queries

- Let $\mathcal{D}$ be a database containing N records. Each record $p \in \mathcal{D}$ is associated with query attribute $A_q(p)$ (age) and a summary attribute $A_s(p)$ (salary).

Summary Queries

- Let $\mathcal{D}$ be a database containing N records. Each record $p \in \mathcal{D}$ is associated with query attribute $A_q(p)$ (age) and a summary attribute $A_s(p)$ (salary).

- A summary query specifies a range constraint $[q_1, q_2]$ on A_q and the database returns a summary on the A_s attribute of all records whose A_q attribute is within the range.

Summary Queries

- Data summarization techniques
  - Heavy hitters (a.k.a. frequent items) [MG 82] [MAA 06] ...
  - Quantiles [MP 80] [GK 01] ...
  - Histograms [PHIJ 96] [JKMPSS 98] [GGIKMS 02] ...
  - Wavelets [MVW 98] [VM 99] [GKMS 01] ...
  - Various sketches ([AMS 99], Count-Min [CM 05], ...)
  - . . .

Summary Queries

- Data summarization techniques
  - Heavy hitters (a.k.a. frequent items) [MG 82] [MAA 06] ...
  - Quantiles [MP 80] [GK 01] ...
  - Histograms [PHIJ 96] [JKMPSS 98] [GGIKMS 02] ...
  - Wavelets [MVW 98] [VM 99] [GKMS 01] ...
  - Various sketches ([AMS 99], Count-Min [CM 05], ...)
  - . . .

- Past research focuses on computing summaries on the whole data set: offline or streaming

Algorithm Problem vs. Data Structure Problem

	The algorithm problem	The data structure problem
Space		
Time		

Algorithm Problem vs. Data Structure Problem

	The algorithm problem	The data structure problem
Space	offline: $O(N)$ streaming: sublinear	$O(N)$: data must be stored
Time		

Algorithm Problem vs. Data Structure Problem

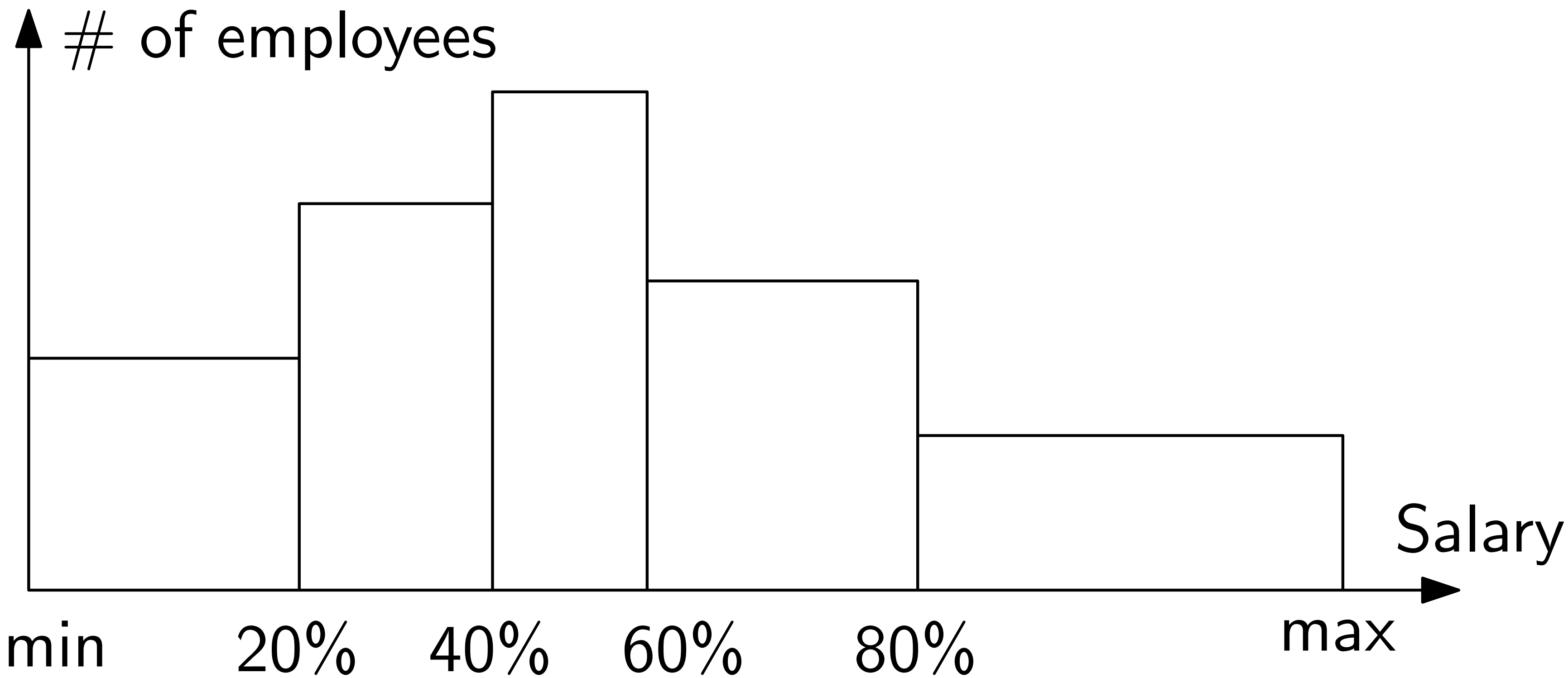
	The algorithm problem	The data structure problem
Space	offline: $O(N)$ streaming: sublinear	$O(N)$: data must be stored
Time	$\tilde{O}(N)$ sublinear when sampling works	preprocessing time: less important
		query time: $O(\log N + s_\varepsilon)$ internal mem $O(\log_B N + s_\varepsilon / B)$ external mem s_ε: summary size B: block size

Quantile Summaries

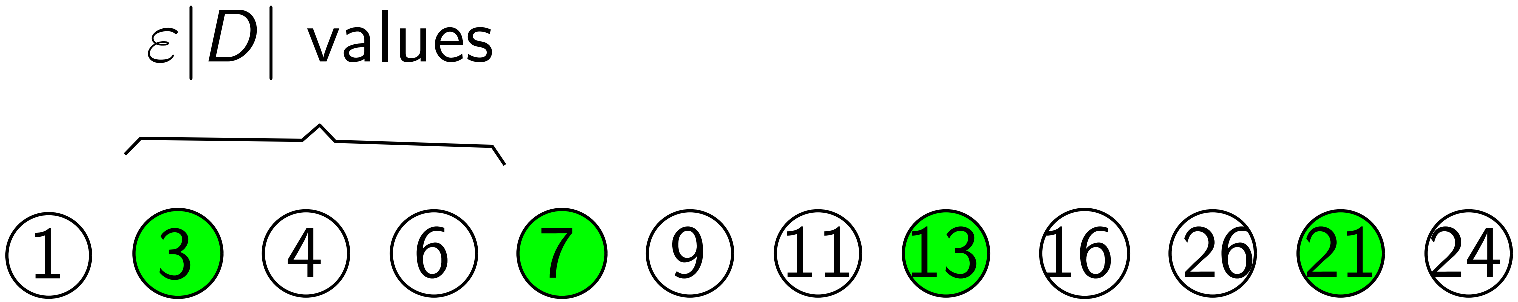
- ϕ-quantile: the value ranked at $\phi|D|$ in D.
- ε-approximate ϕ-quantile: any value whose rank is between $[(\phi - \varepsilon)|D|, (\phi + \varepsilon)|D|]$.
- Quantile summary: for any $0 < \phi < 1$, an ε-approximate ϕ-quantile can be extracted.

Quantile Summaries

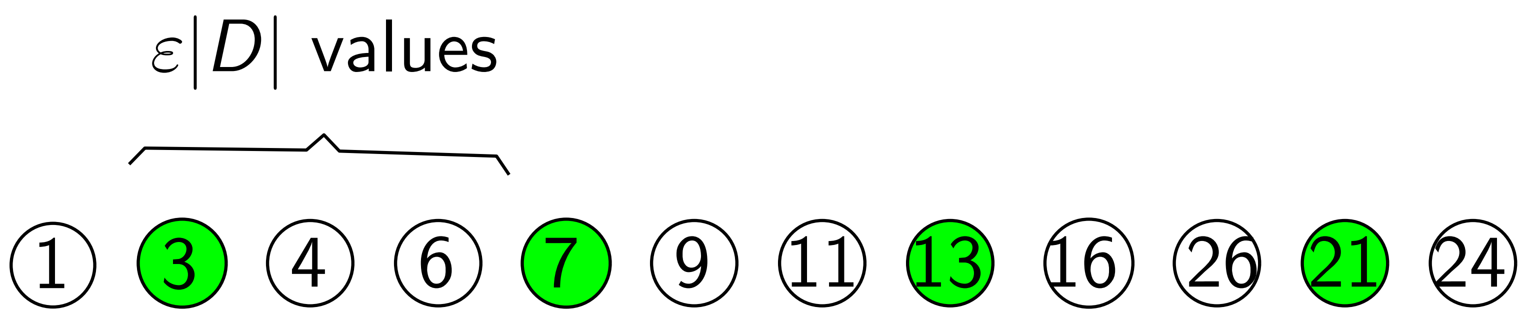
- ϕ-quantile: the value ranked at $\phi|D|$ in D.
- ε-approximate ϕ-quantile: any value whose rank is between $[(\phi-\varepsilon)|D|, (\phi+\varepsilon)|D|]$.
- Quantile summary: for any $0 < \phi < 1$, an ε-approximate ϕ-quantile can be extracted.



Quantile Summaries



Quantile Summaries



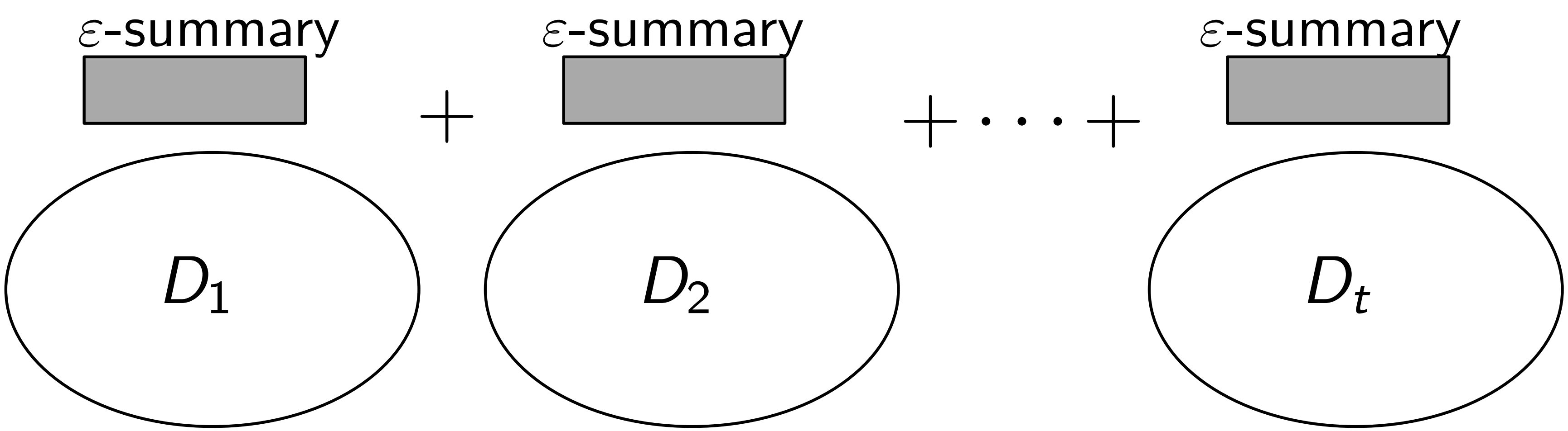
Size: $s_\varepsilon = \Theta(1/\varepsilon)$; Error: $\varepsilon|D|$

A Baseline Solution

- Decomposable summaries

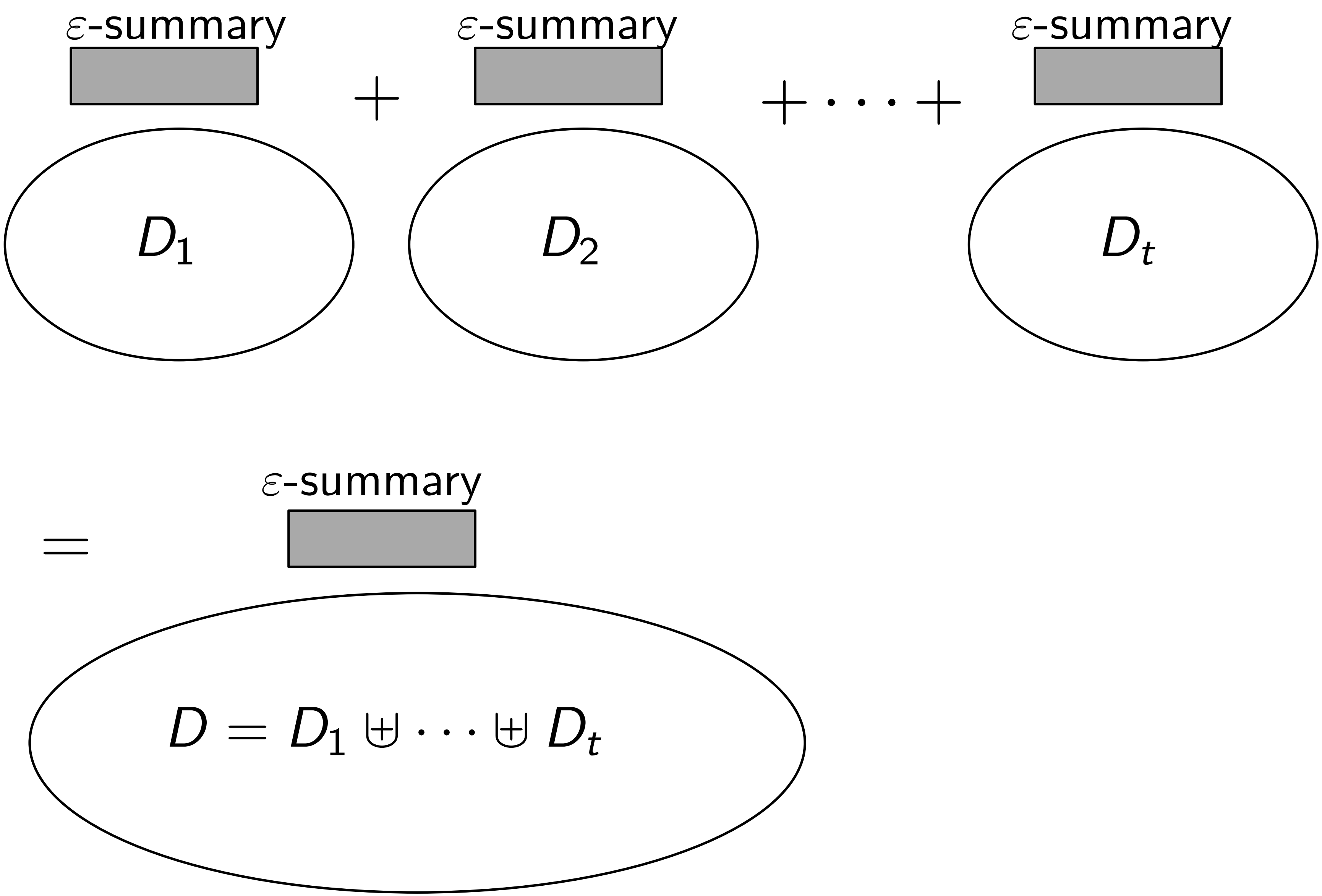
A Baseline Solution

- Decomposable summaries



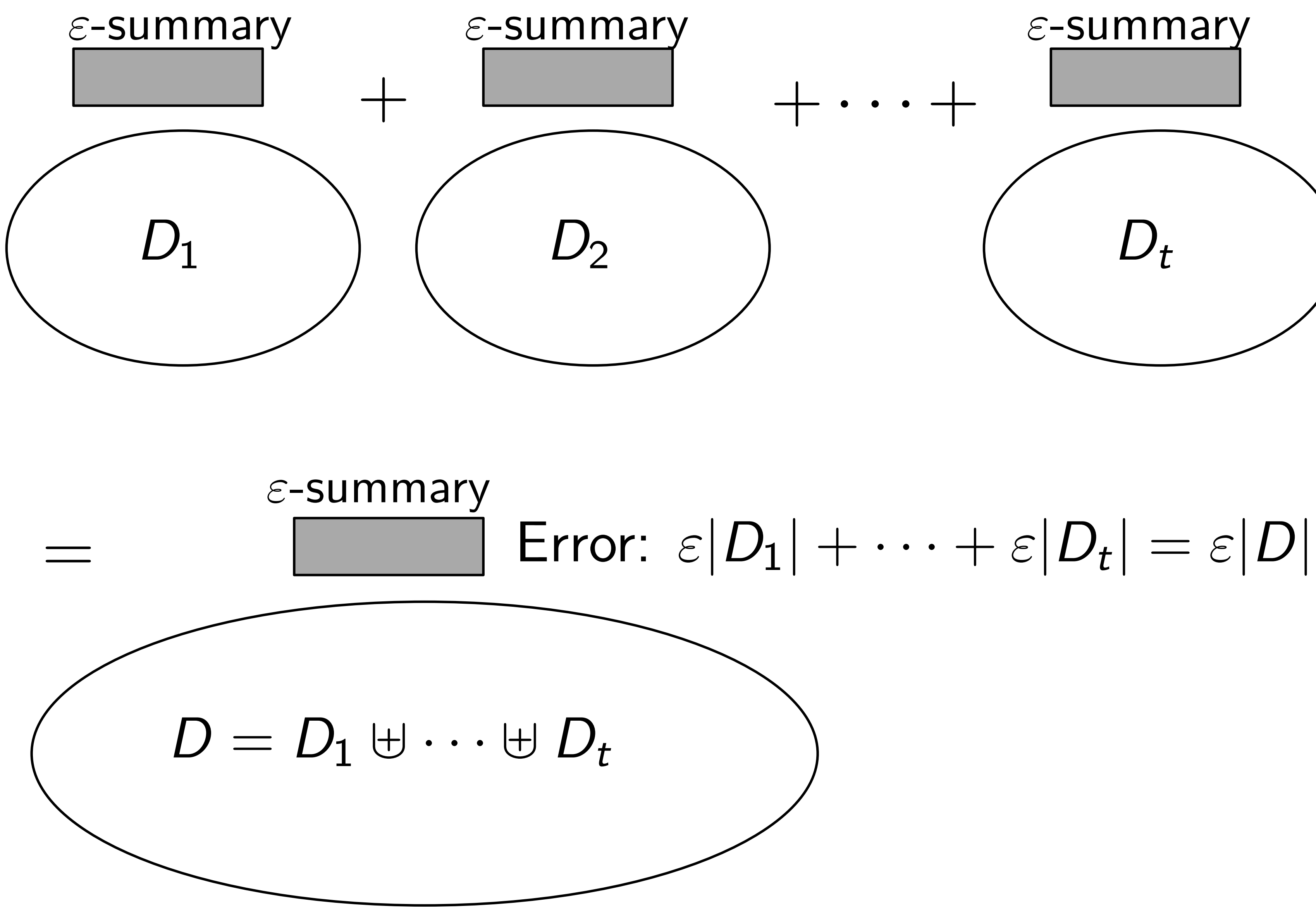
A Baseline Solution

- Decomposable summaries

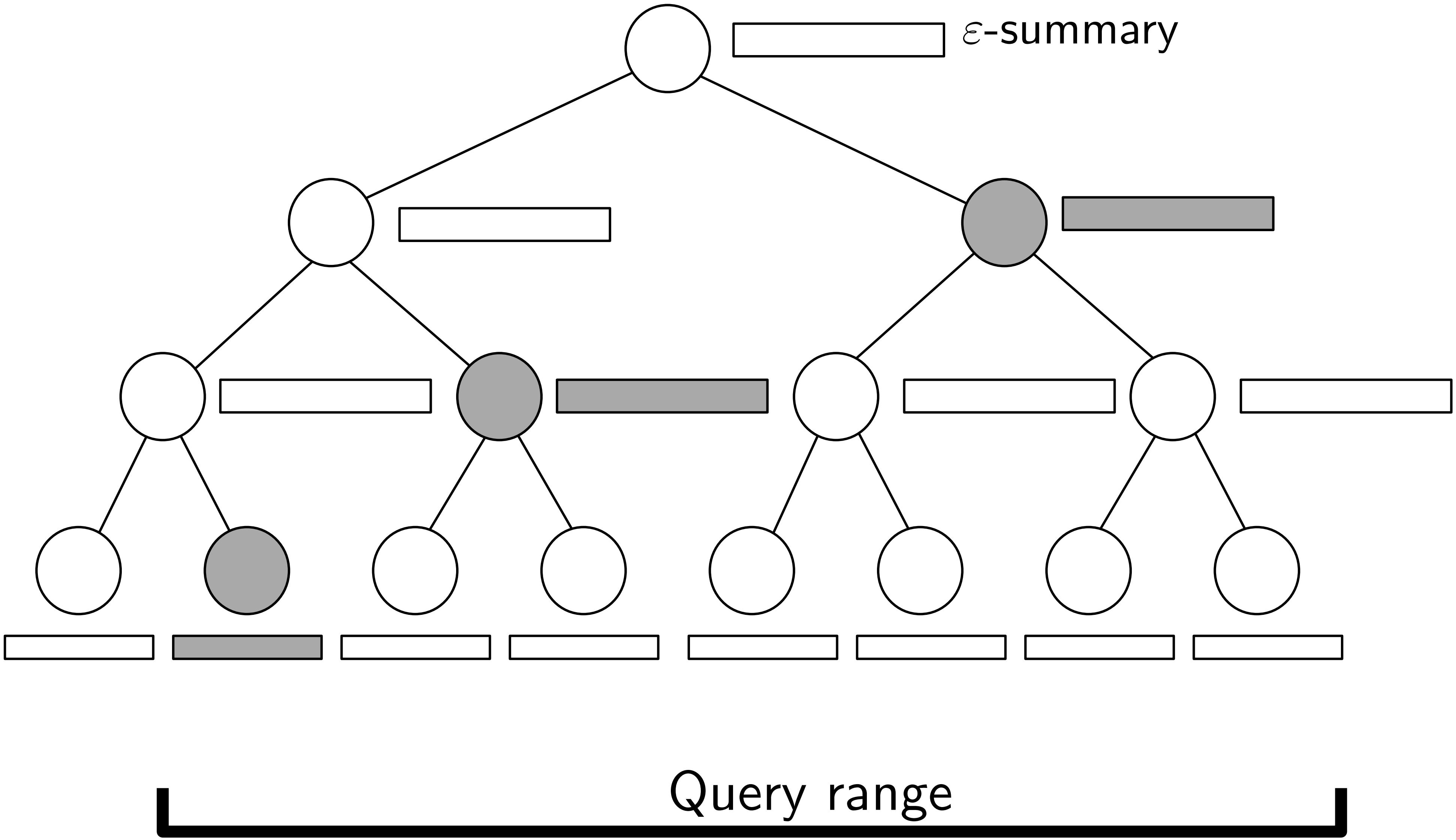


A Baseline Solution

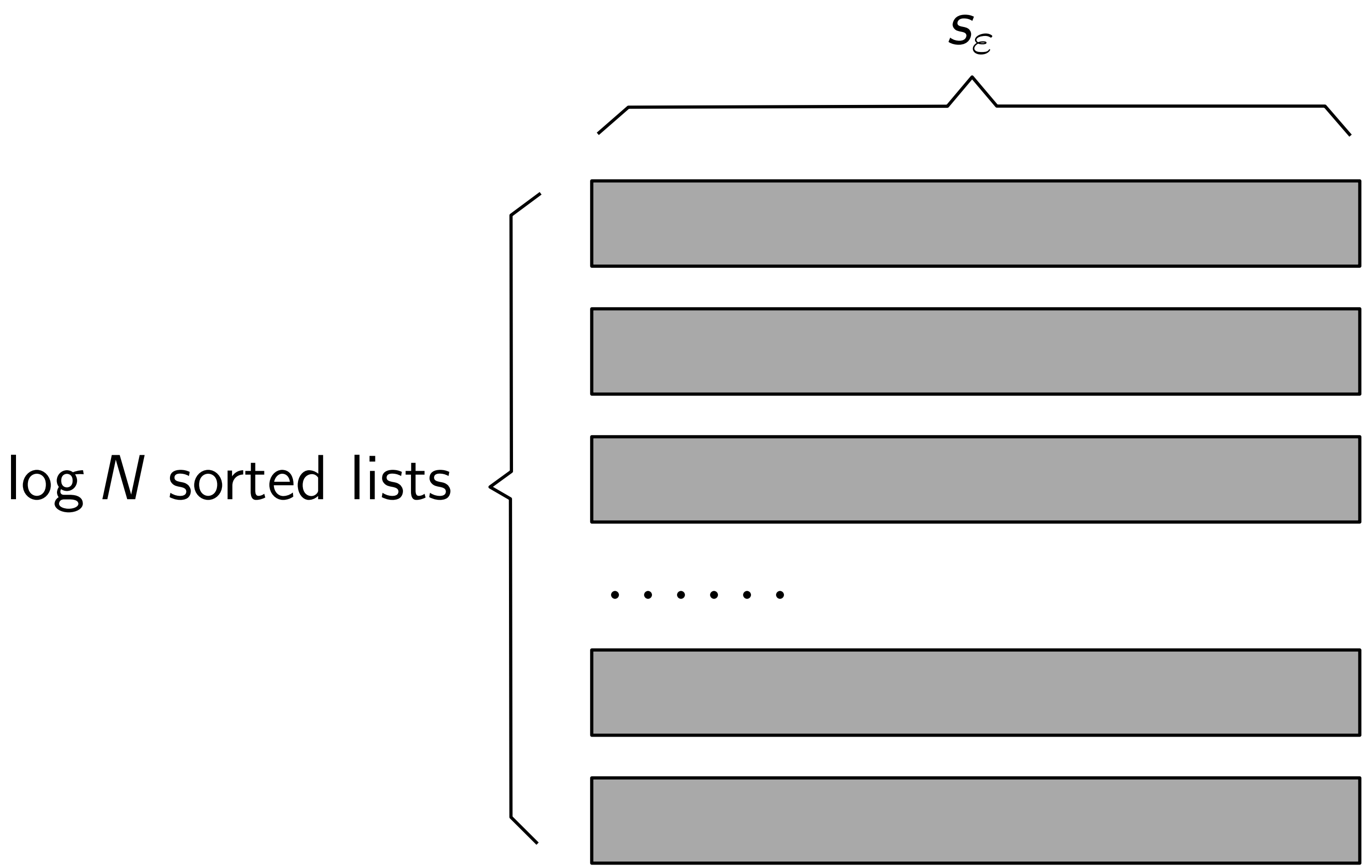
- Decomposable summaries



A Baseline Solution



Query Cost



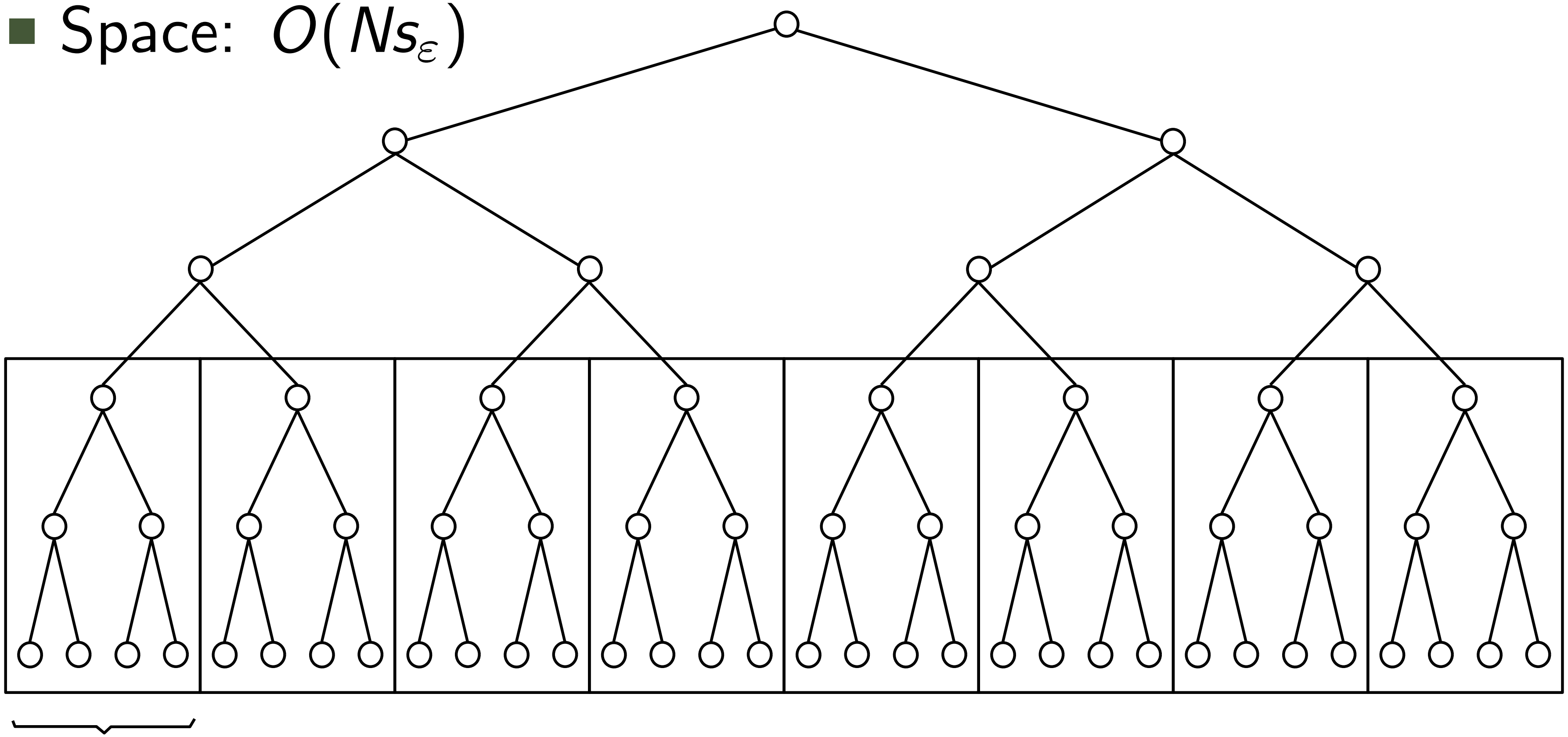
log N-way merging: $O(s_\varepsilon \log N \log\log N)$

A Baseline Solution

- Internal memory
  - Query time: $O(s_\varepsilon \log N \log\log N)$
  - Space: $O(Ns_\varepsilon)$

A Baseline Solution

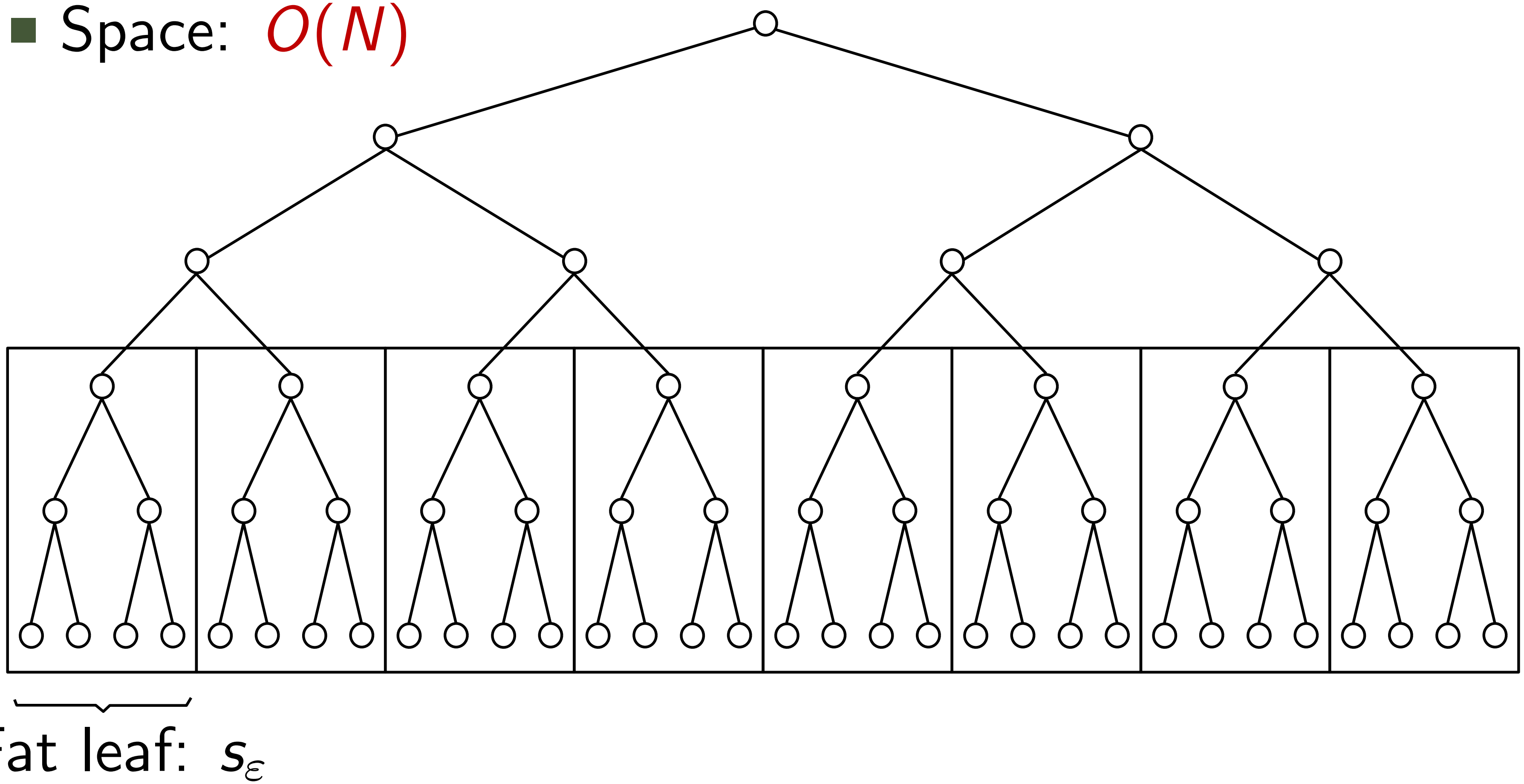
- Internal memory
  - Query time: $O(s_\varepsilon \log N \log\log N)$
  - Space: $O(Ns_\varepsilon)$



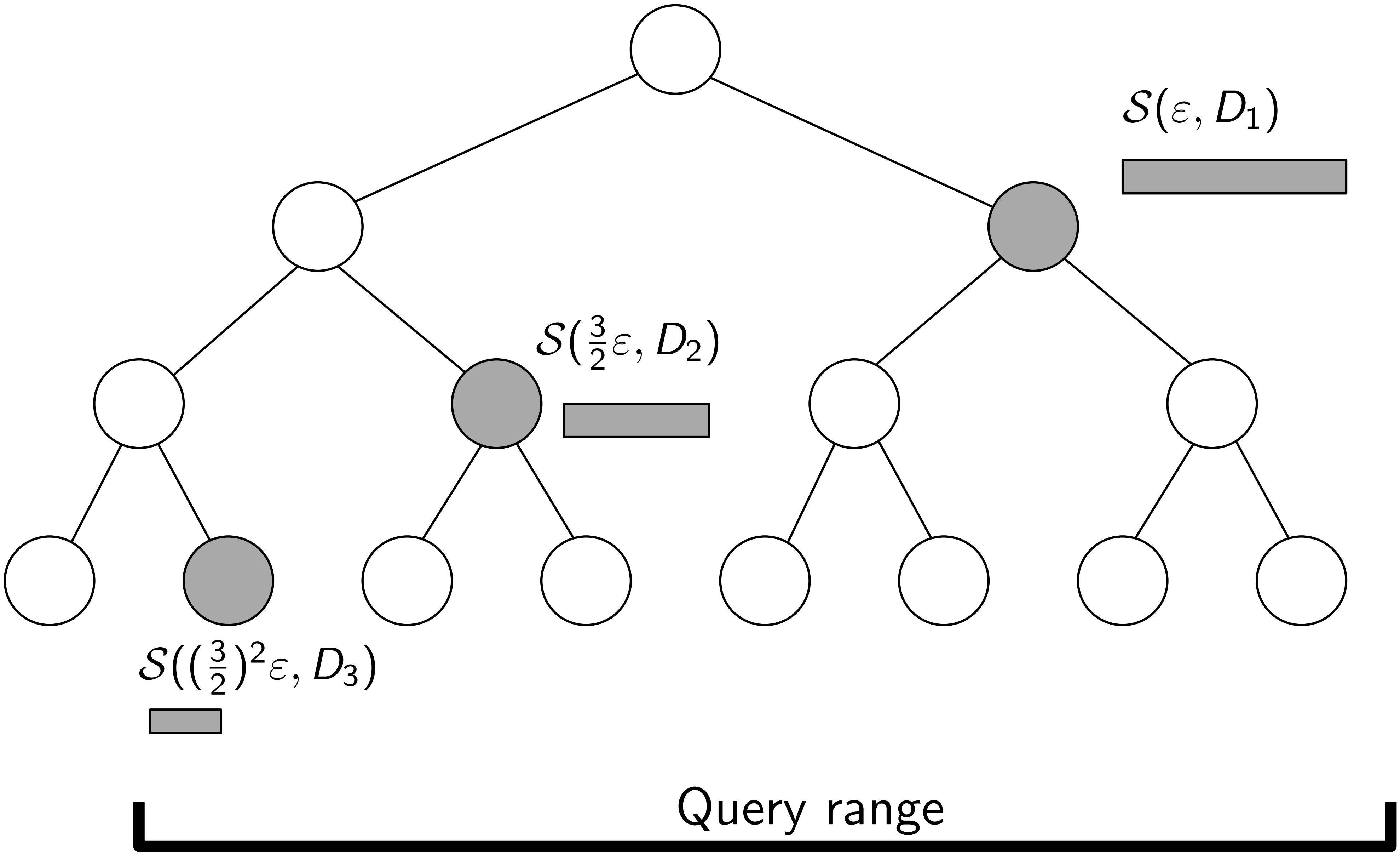
Fat leaf: s_ε

A Baseline Solution

- Internal memory
  - Query time: $O(s_\varepsilon \log N \log \log N)$
  - Space: $O(N)$



Optimal Data Structure



Optimal Data Structure

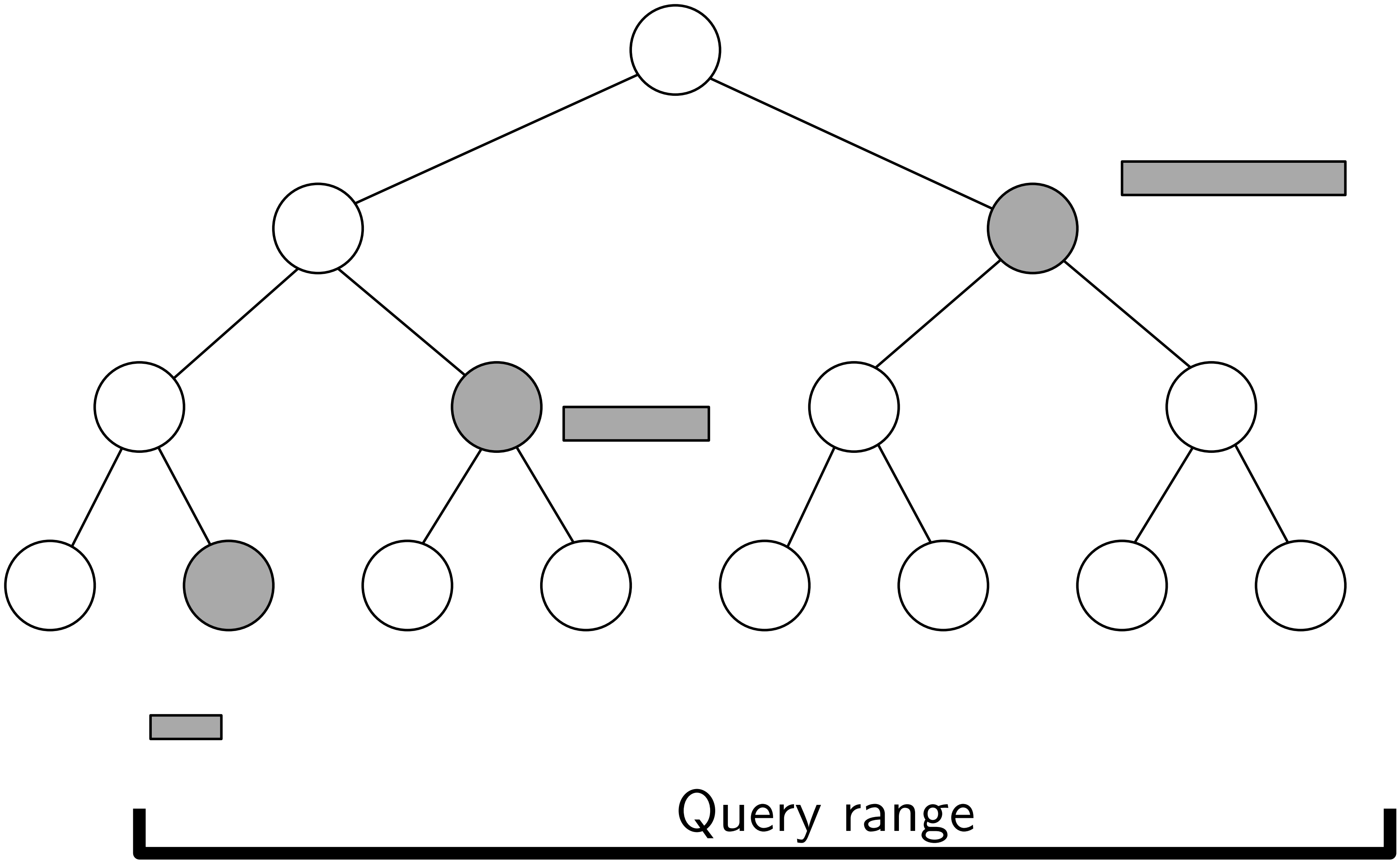
- Quantile summary
  - $\mathcal{S}(\varepsilon, D)$: An ε-quantile summary for data set D.
  - Size: $\Theta(1/\varepsilon)$; Error: $\varepsilon|D|$.

Optimal Data Structure

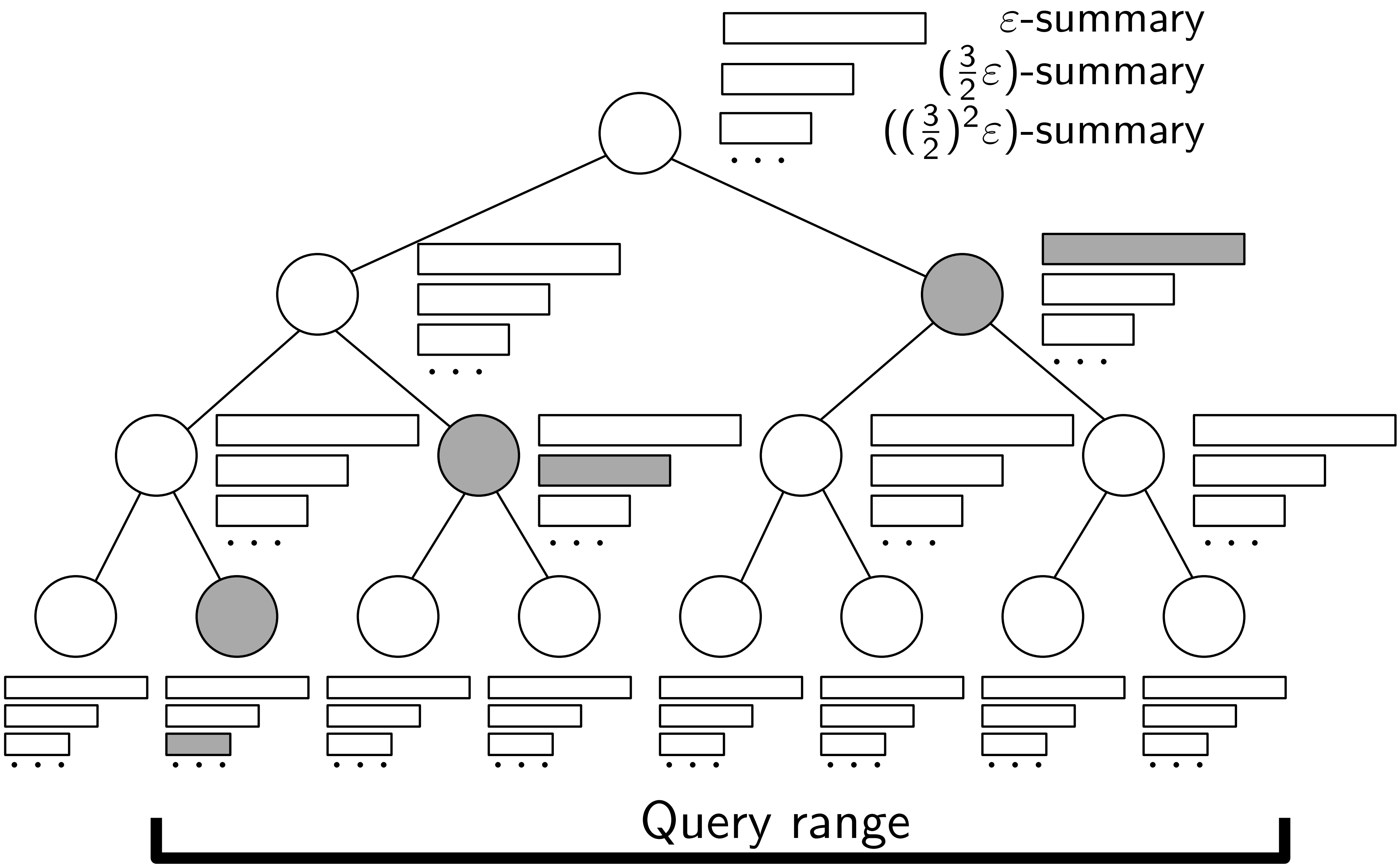
- Quantile summary
  - $\mathcal{S}(\varepsilon, D)$: An ε-quantile summary for data set D.
  - Size: $\Theta(1/\varepsilon)$; Error: $\varepsilon|D|$.

Data set	Data size	Error param.	Summary size	Absolute error
D_1	k	ε	$\frac{1}{\varepsilon}$	εk
D_2	$\frac{k}{2}$	$\frac{3}{2}\varepsilon$	$\frac{2}{3}\frac{1}{\varepsilon}$	$\frac{3}{4}\varepsilon k$
D_3	$\frac{k}{4}$	$\left(\frac{3}{2}\right)^2 \varepsilon$	$\left(\frac{2}{3}\right)^2 \frac{1}{\varepsilon}$	$\left(\frac{3}{4}\right)^2 \varepsilon k$
$\ldots$				
D_t	$\frac{k}{2^{t-1}}$	$\left(\frac{3}{2}\right)^{t-1} \varepsilon$	$\left(\frac{2}{3}\right)^{t-1} \frac{1}{\varepsilon}$	$\left(\frac{3}{4}\right)^{t-1} \varepsilon k$
D	$\Theta(k)$		$O(\frac{1}{\varepsilon})$	$O(\varepsilon k)$

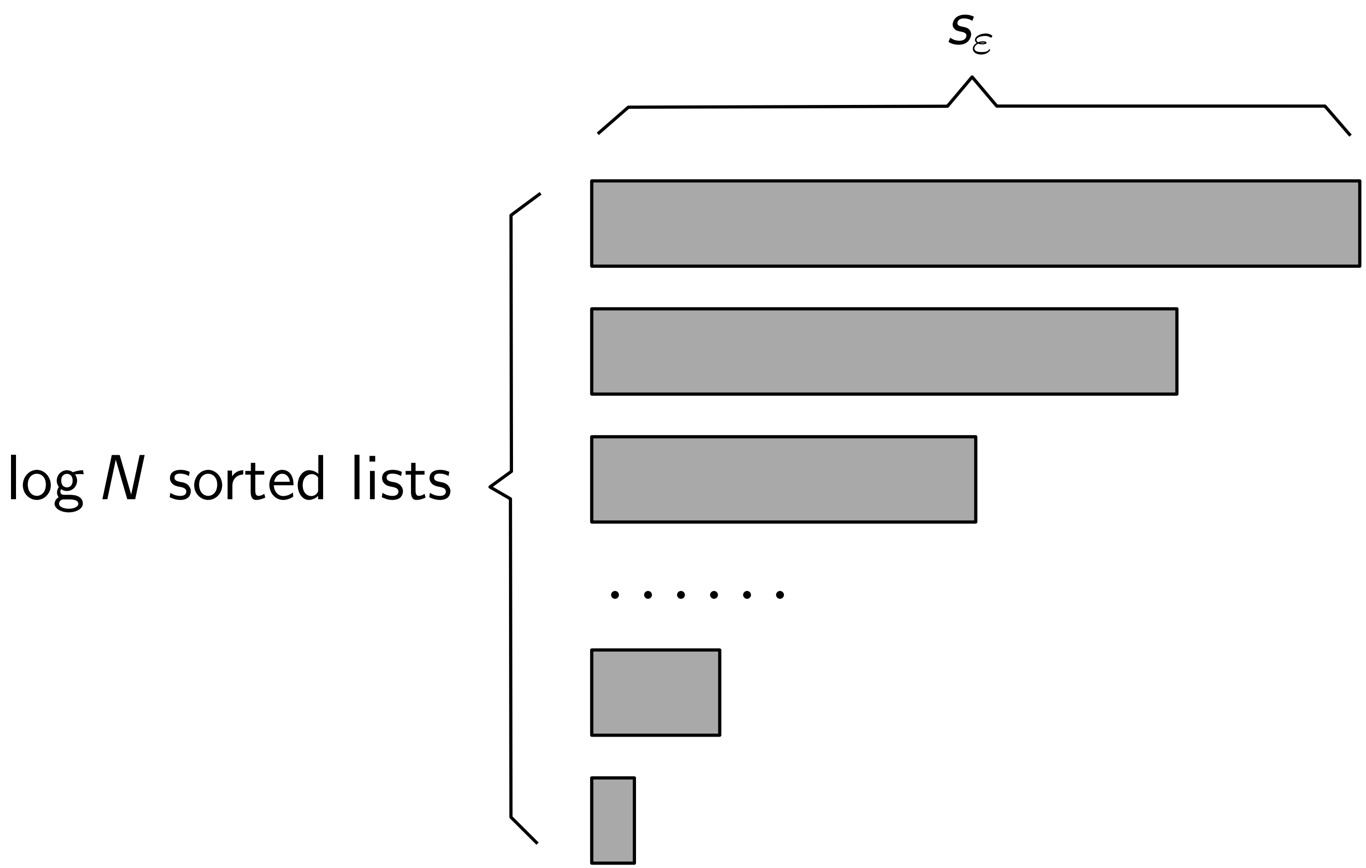
Optimal Data Structure



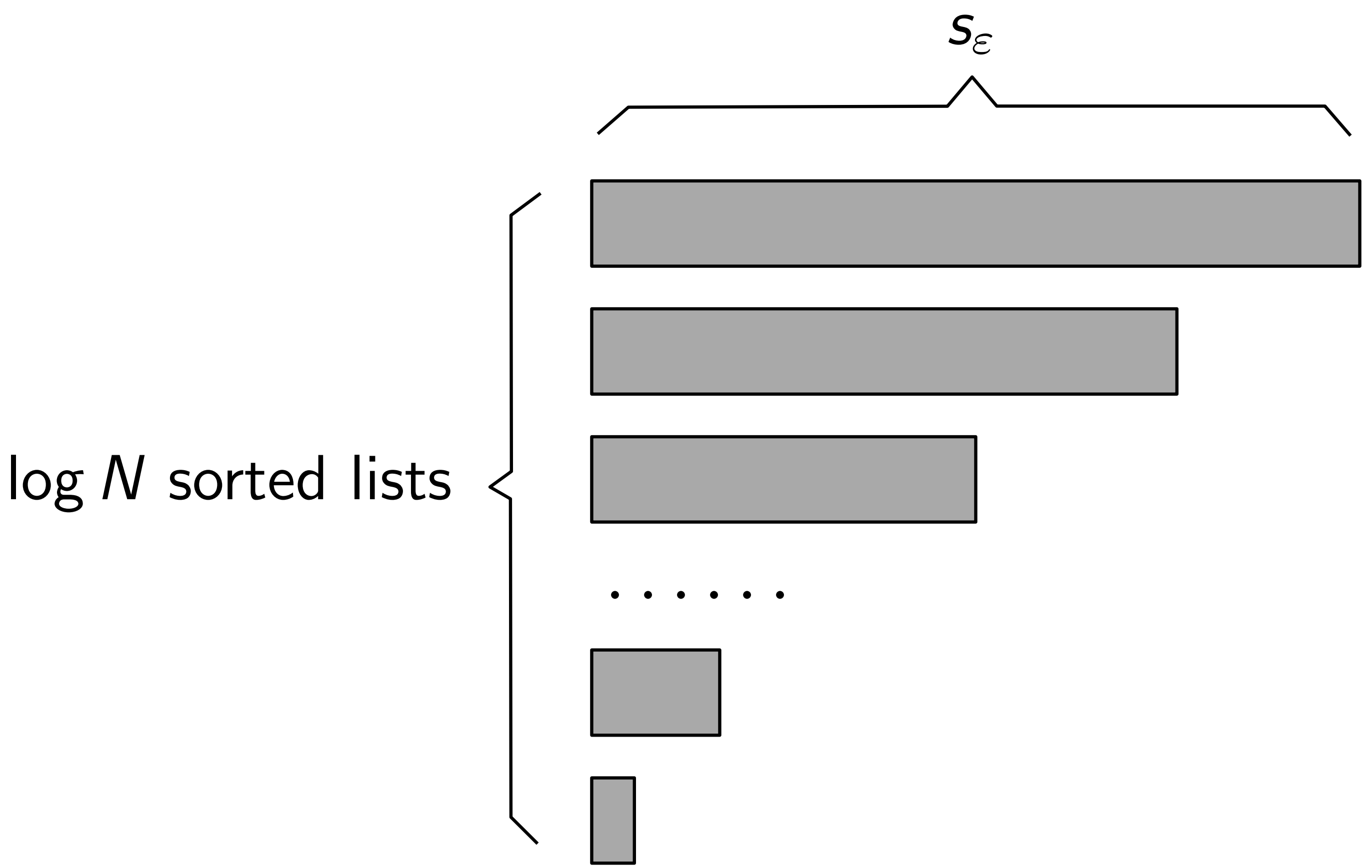
Optimal Data Structure



Query Cost

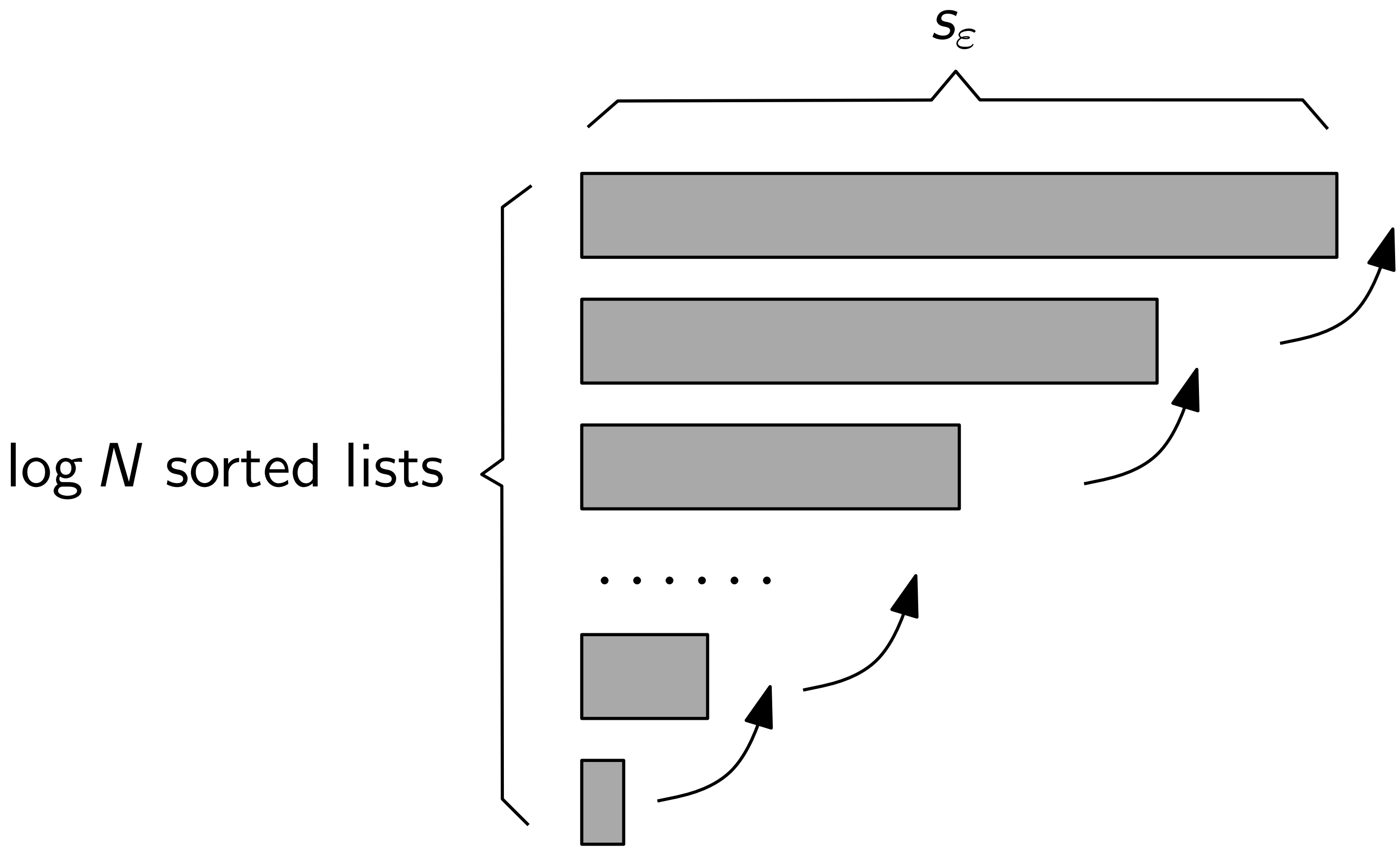


Query Cost

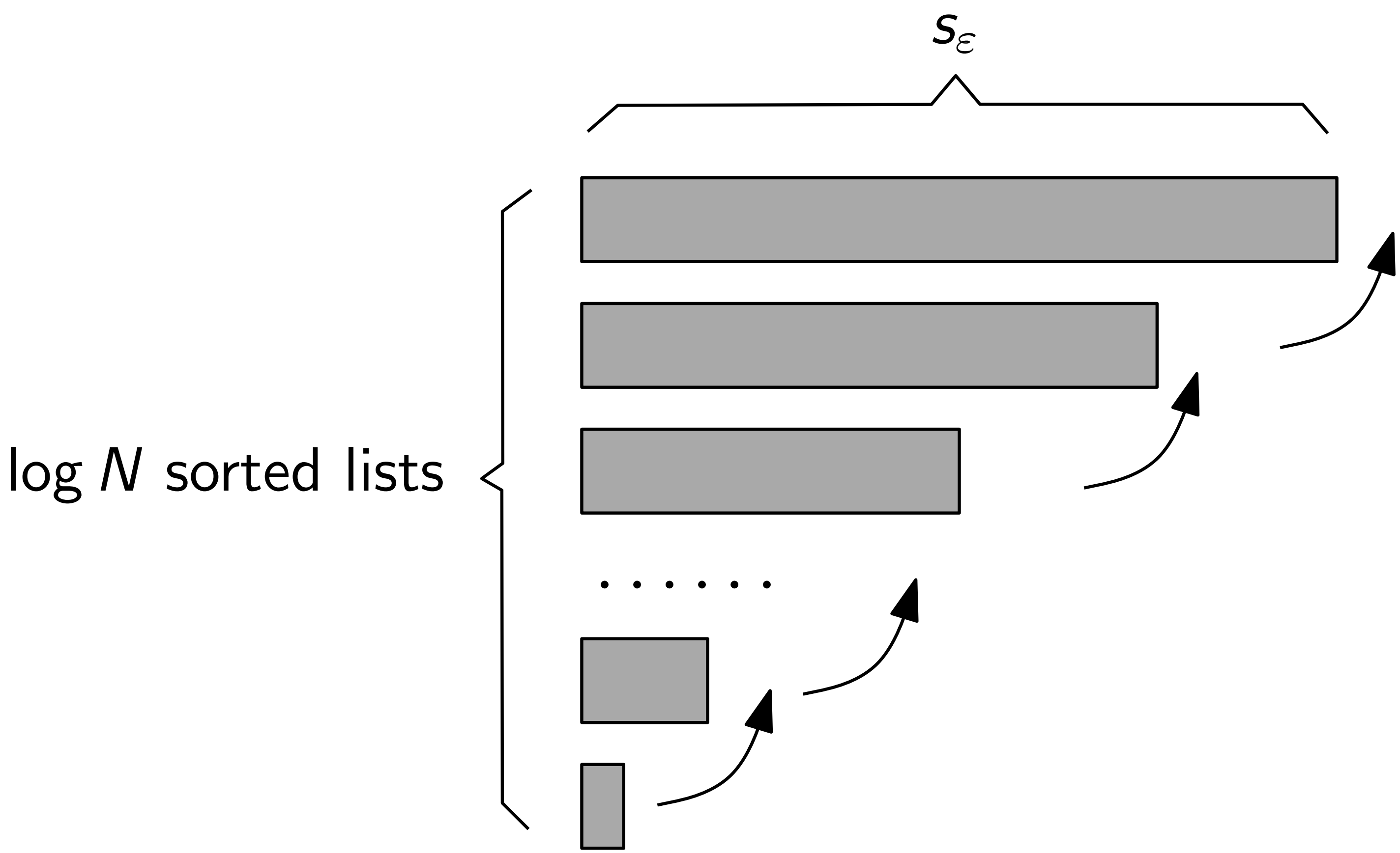


log N-way merging: $\Theta(s_\varepsilon \log\log N)$

Query Cost



Query Cost



Bottom-up two-way merging: $O(s_\varepsilon)$

α-Exponentially Decomposable

- Multisets $D_1, \ldots, D_t$ with $F_1(D_i) \leq \alpha^{i-1} F_1(D_1)$, $\exists$ constant c, s.t. given $\mathcal{S}(\varepsilon, D_1), \mathcal{S}(c\varepsilon, D_2) \ldots, \mathcal{S}(c^{t-1}\varepsilon, D_t)$:
  - We can construct an $O(\varepsilon)$-summary for $D_1 \uplus \cdots \uplus D_t$.
  - The total size of $\mathcal{S}(\varepsilon, D_1), \ldots, \mathcal{S}(c^{t-1}\varepsilon, D_t)$ is $O(s_\varepsilon)$ and they can be combined in $O(s_\varepsilon)$ time.
  - The total size of $\mathcal{S}(\varepsilon, D), \ldots, \mathcal{S}(c^{t-1}\varepsilon, D)$ is $O(s_\varepsilon)$.

α-Exponentially Decomposable

- Multisets $D_1, \ldots, D_t$ with $F_1(D_i) \le \alpha^{i-1} F_1(D_1)$, $\exists$ constant c, s.t. given $\mathcal{S}(\varepsilon, D_1), \mathcal{S}(c\varepsilon, D_2) \ldots, \mathcal{S}(c^{t-1}\varepsilon, D_t)$:
  - We can construct an $O(\varepsilon)$-summary for $D_1 \uplus \cdots \uplus D_t$.
  - The total size of $\mathcal{S}(\varepsilon, D_1), \ldots, \mathcal{S}(c^{t-1}\varepsilon, D_t)$ is $O(s_\varepsilon)$ and they can be combined in $O(s_\varepsilon)$ time.
  - The total size of $\mathcal{S}(\varepsilon, D), \ldots, \mathcal{S}(c^{t-1}\varepsilon, D)$ is $O(s_\varepsilon)$.

Theorem

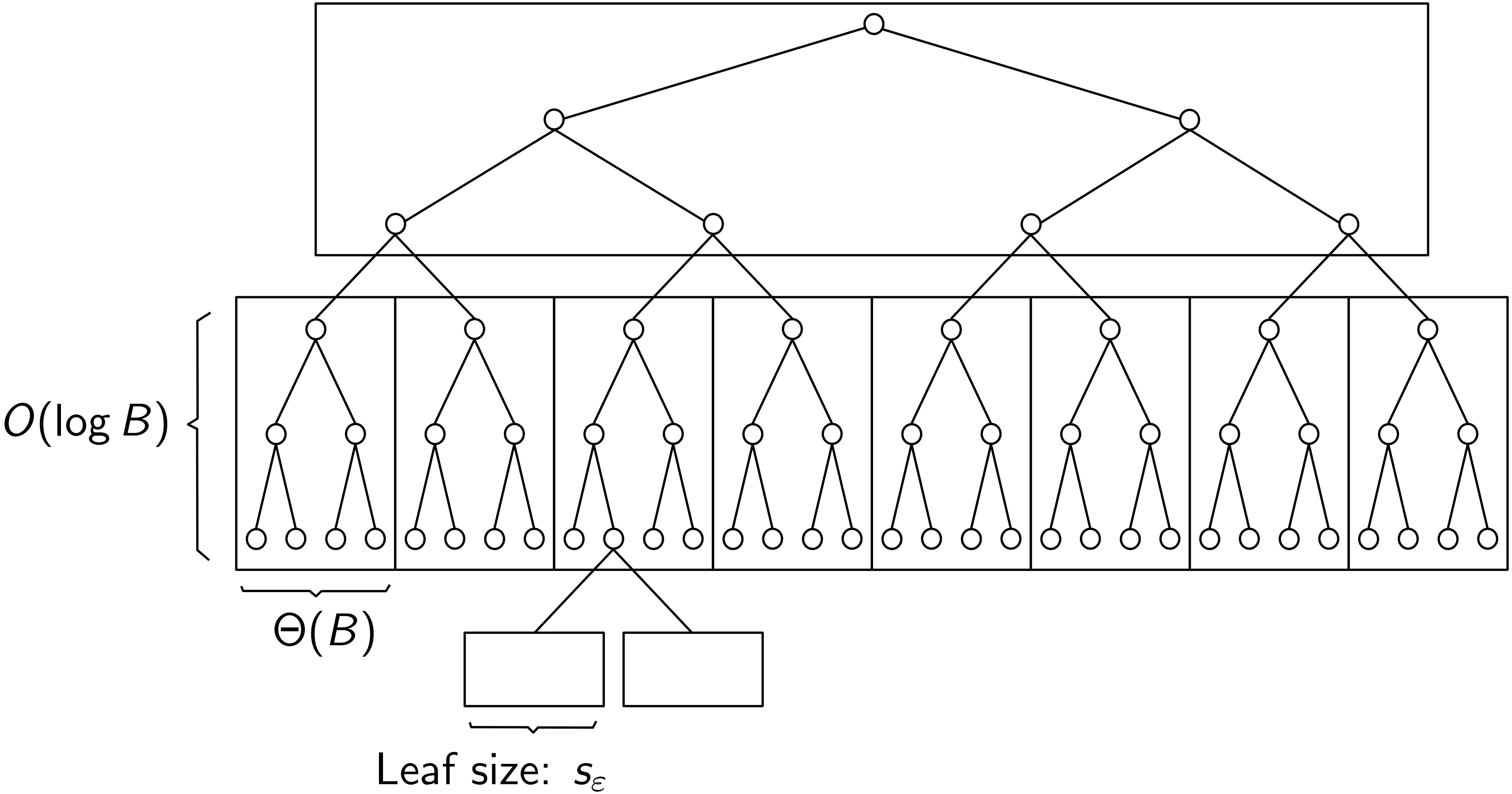
For any $(1/2)$-exponentially decomposable summary, a database $\mathfrak{D}$ of N records can be stored in an internal memory structure of linear size so that a summary query can be answered in $O(\log N + s_\varepsilon)$ time.

Optimal Data Structure - External Memory

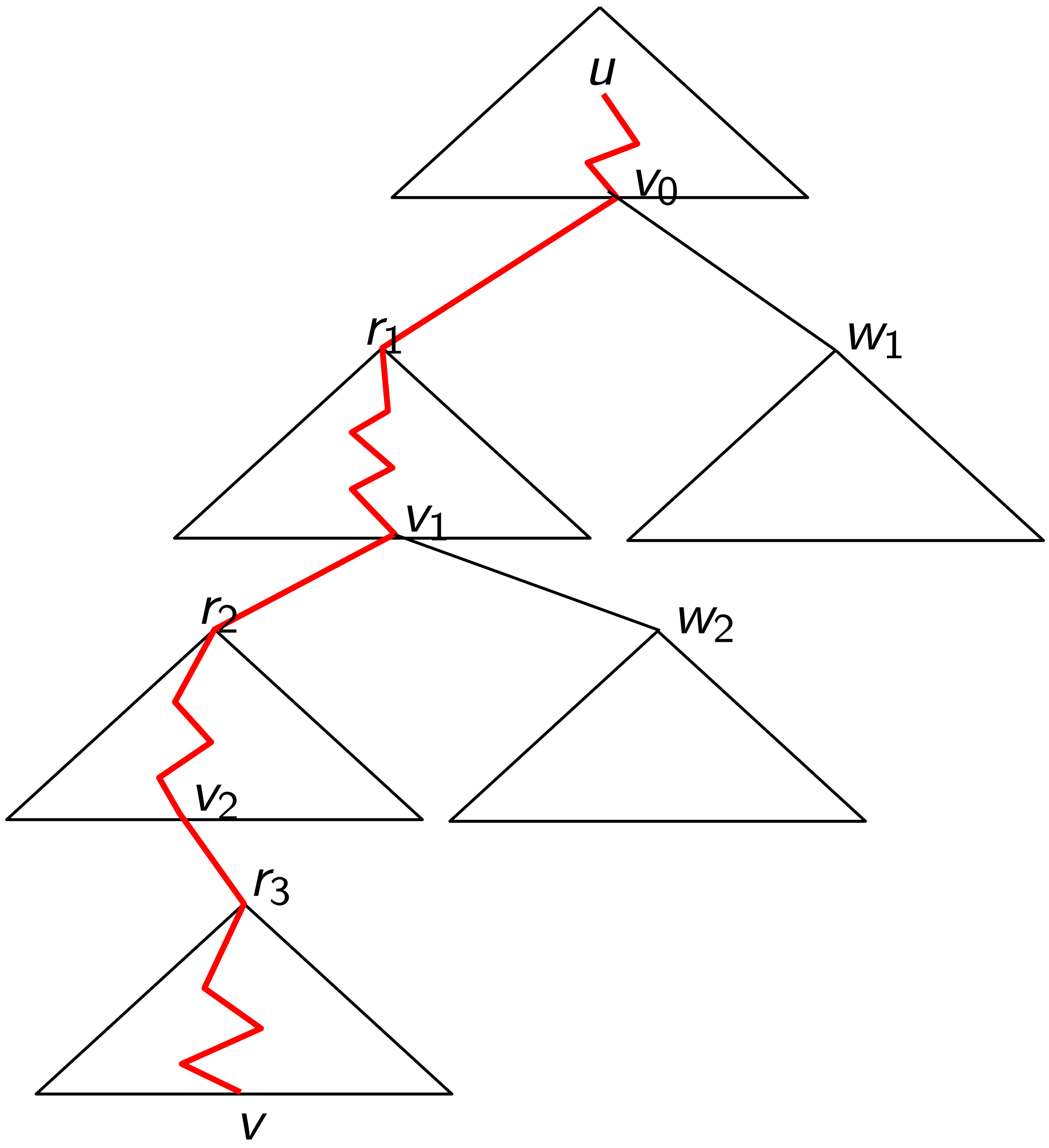
- Standard B-tree blocking with fat leaves

Optimal Data Structure - External Memory

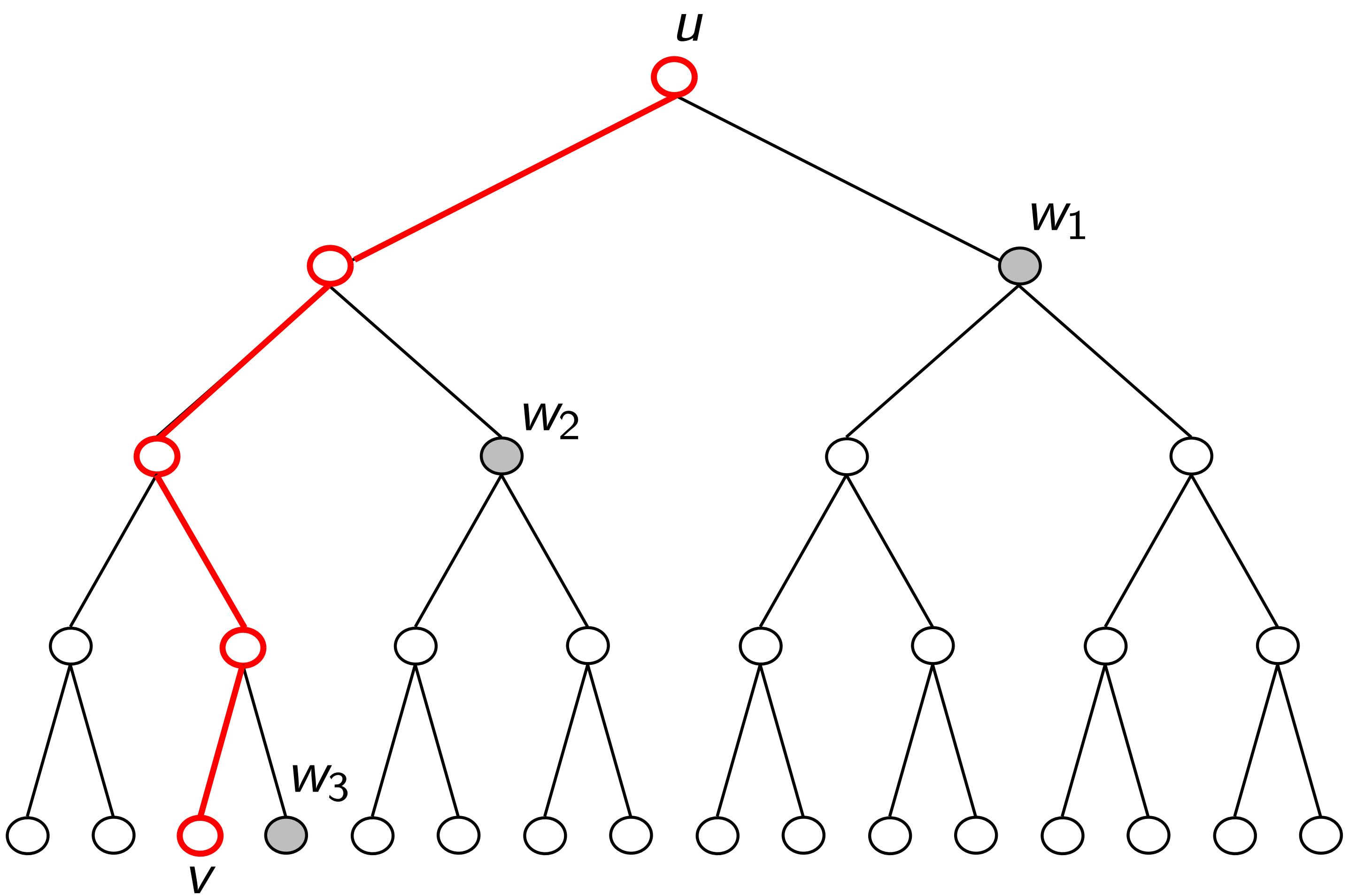
- Standard B-tree blocking with fat leaves



Query Path

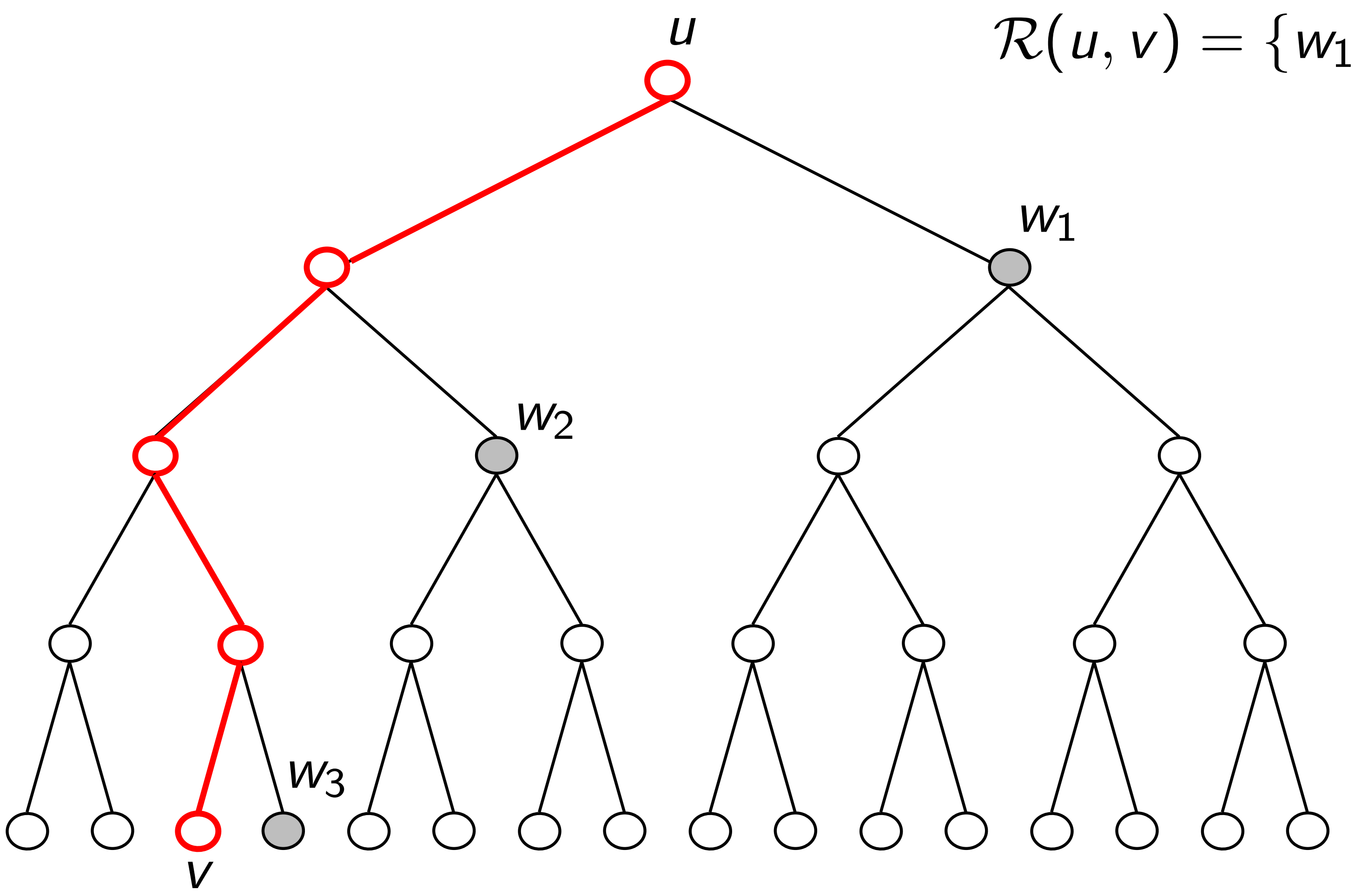


Summary Set

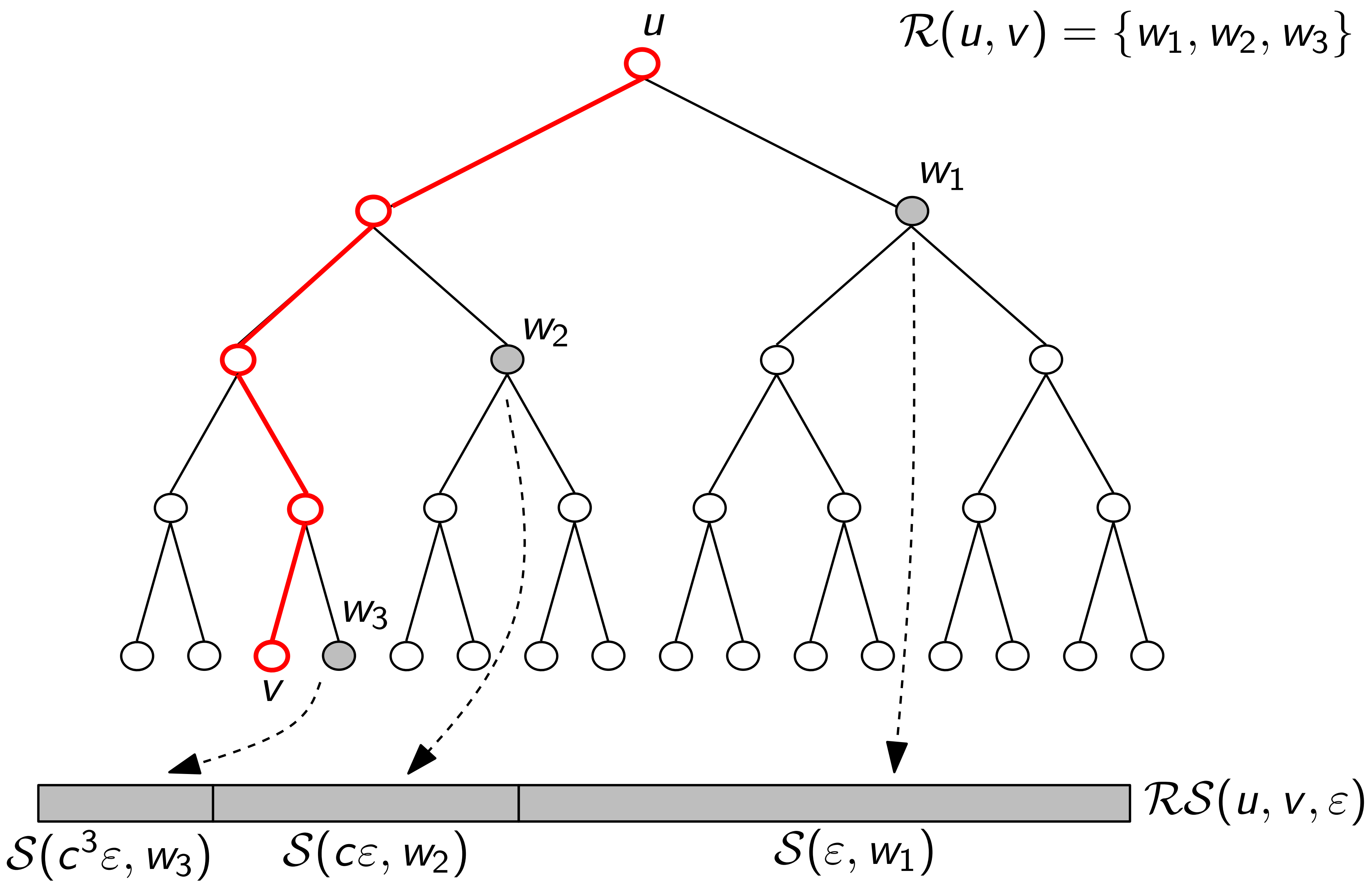


Summary Set

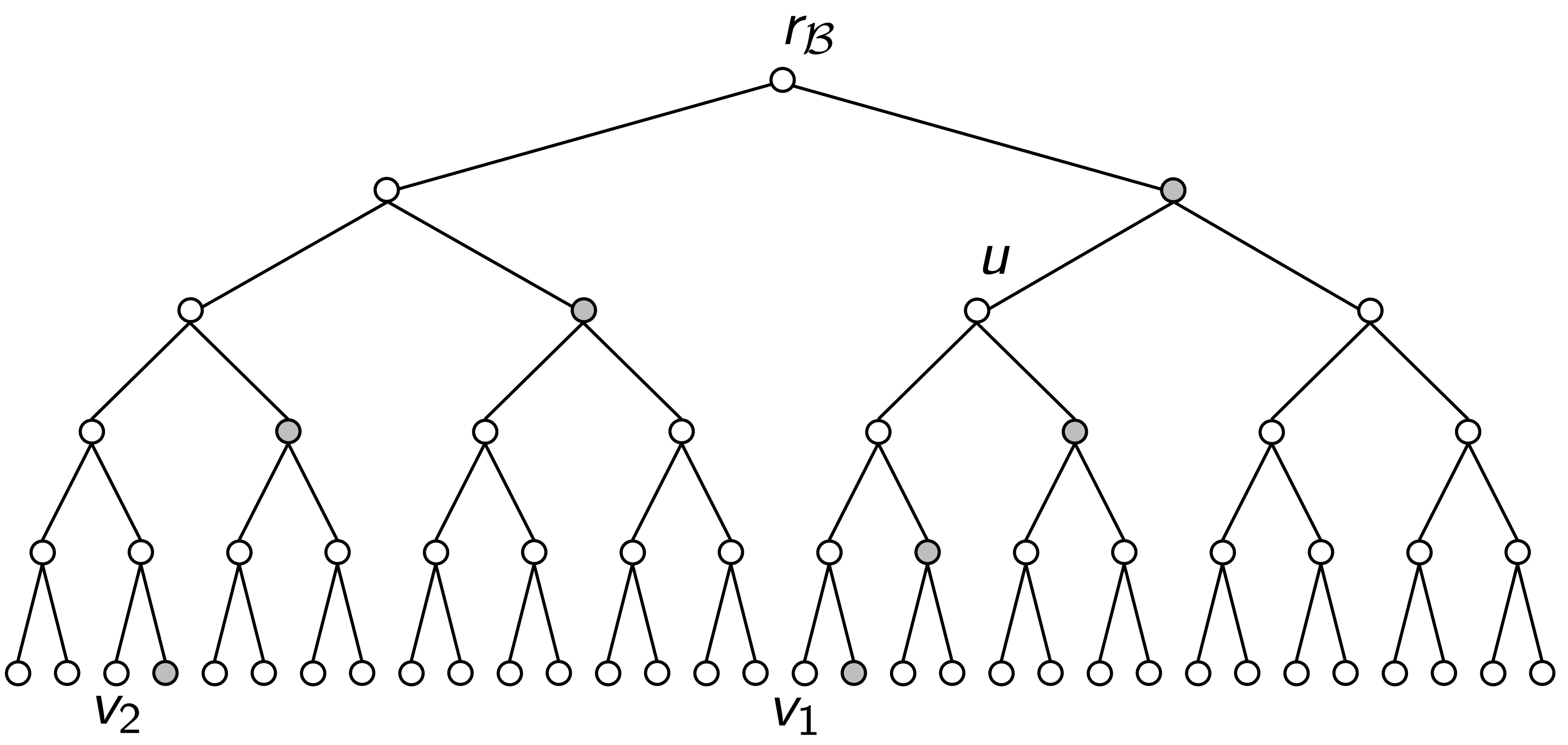
$\mathcal{R}(u, v) = \{w_1, w_2, w_3\}$



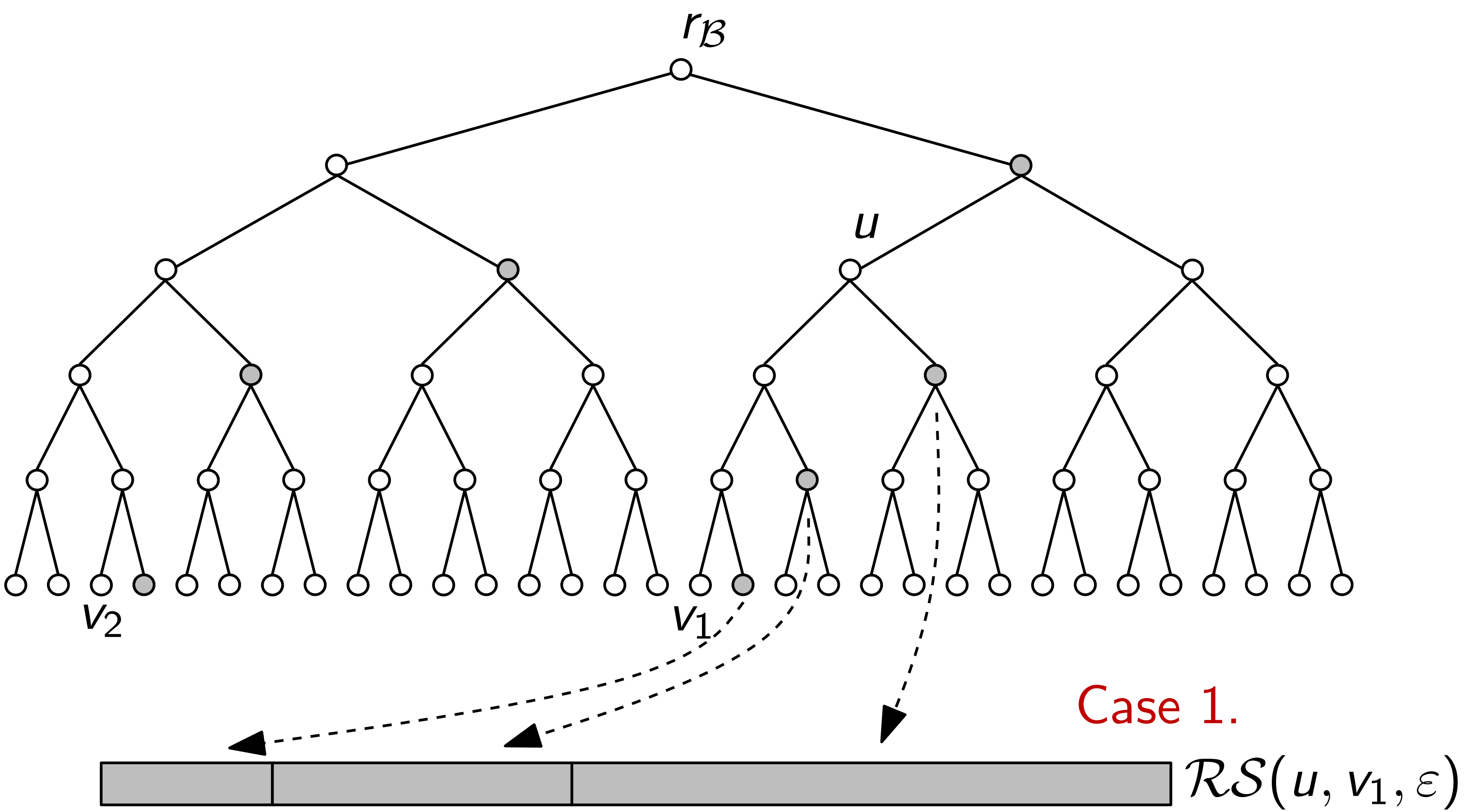
Summary Set


u
$\mathcal{R}(u, v) = \{w_1, w_2, w_3\}$
w_1
w_2
w_3
v
$\mathcal{RS}(u, v, \varepsilon)$
$\mathcal{S}(c^3\varepsilon, w_3)$
$\mathcal{S}(c\varepsilon, w_2)$
$\mathcal{S}(\varepsilon, w_1)$

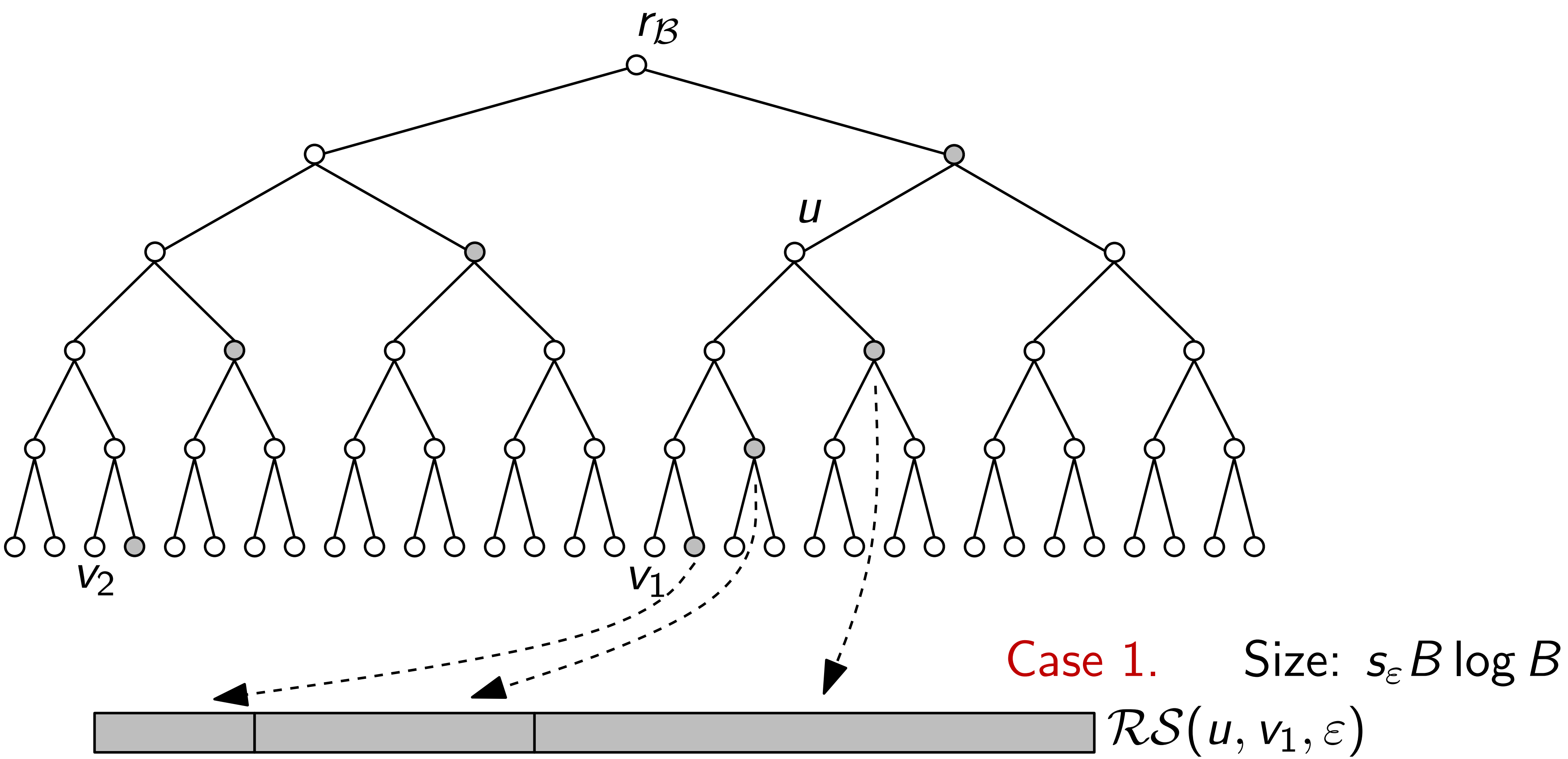
Focus on a Block



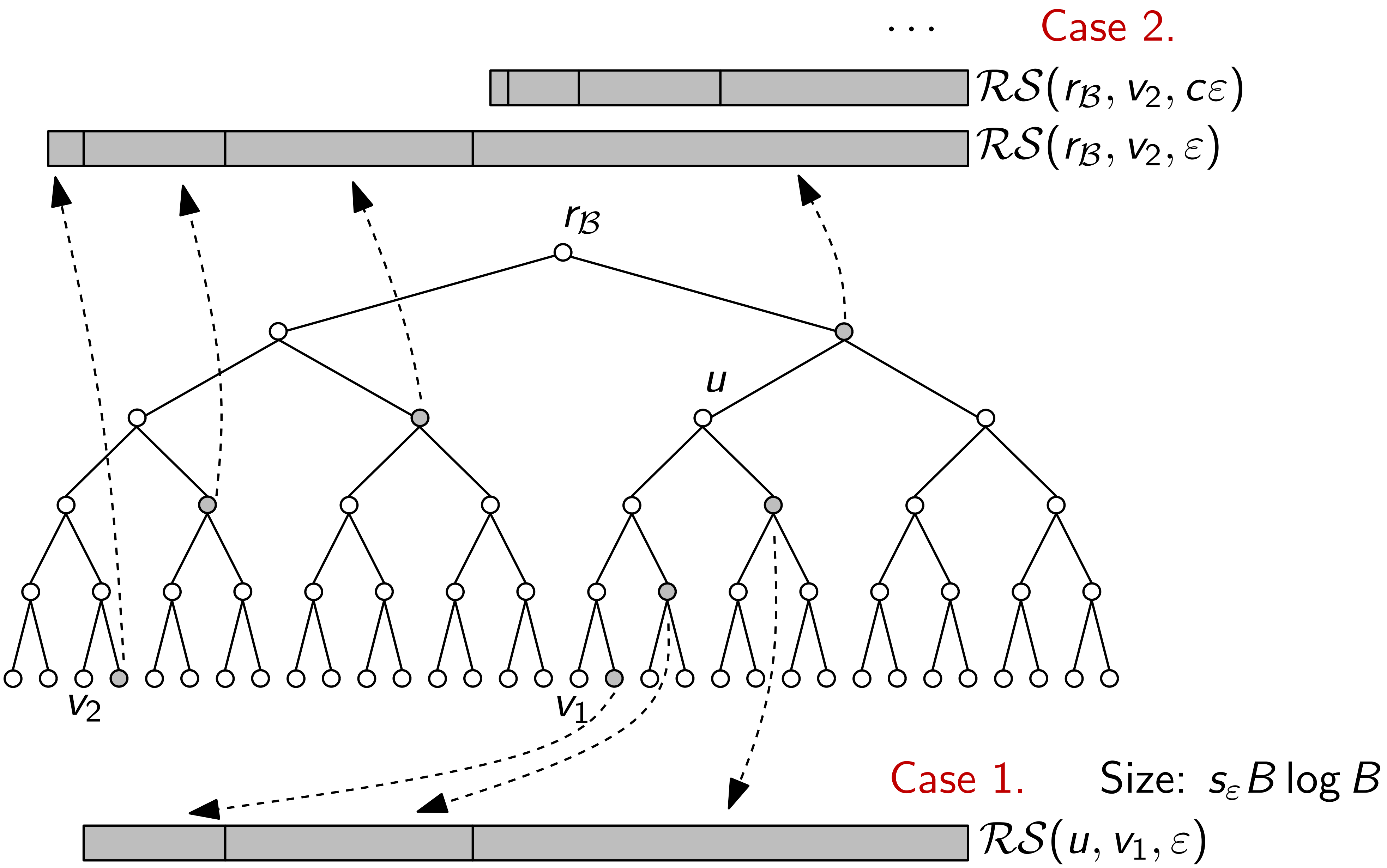
Focus on a Block



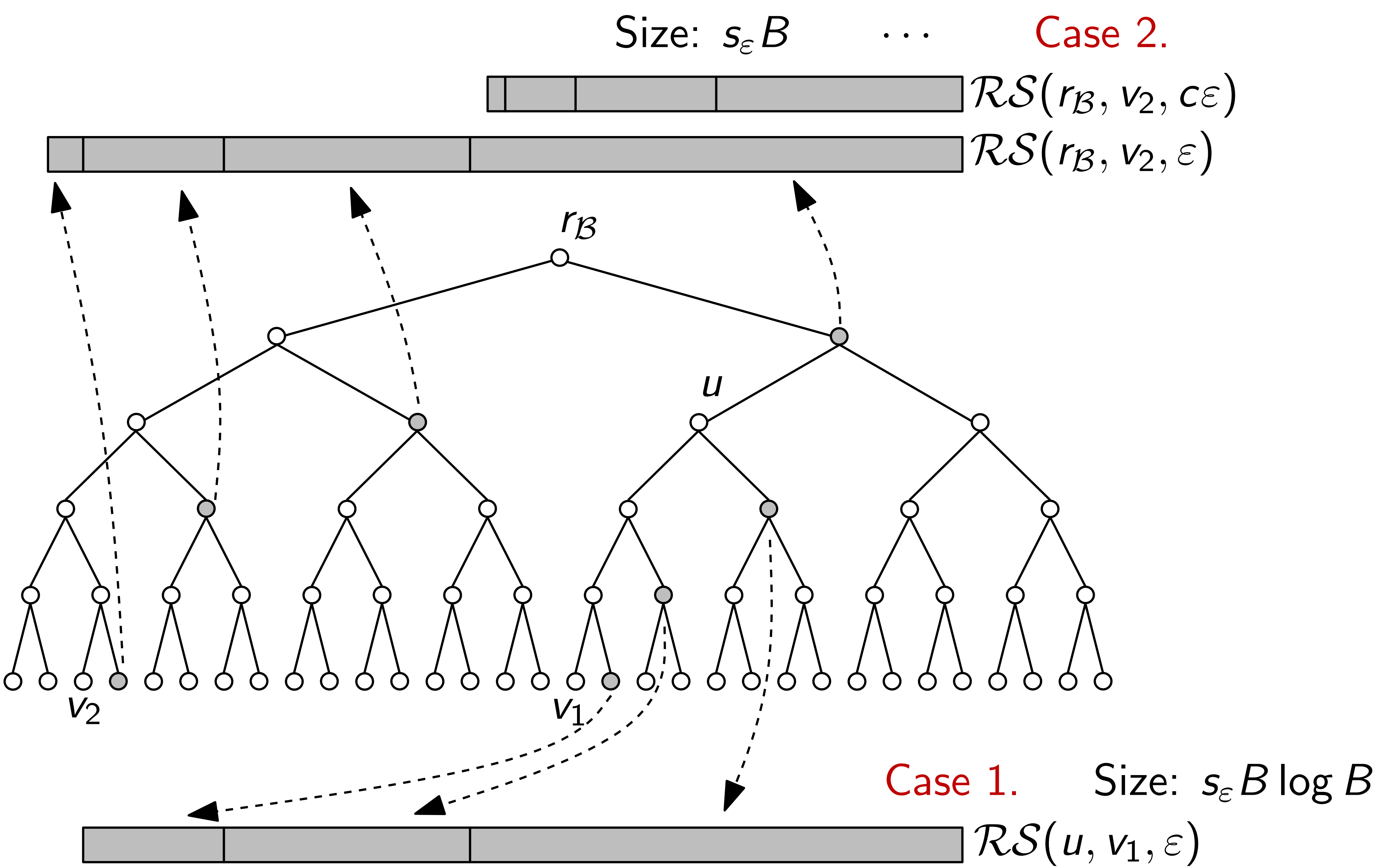
Focus on a Block



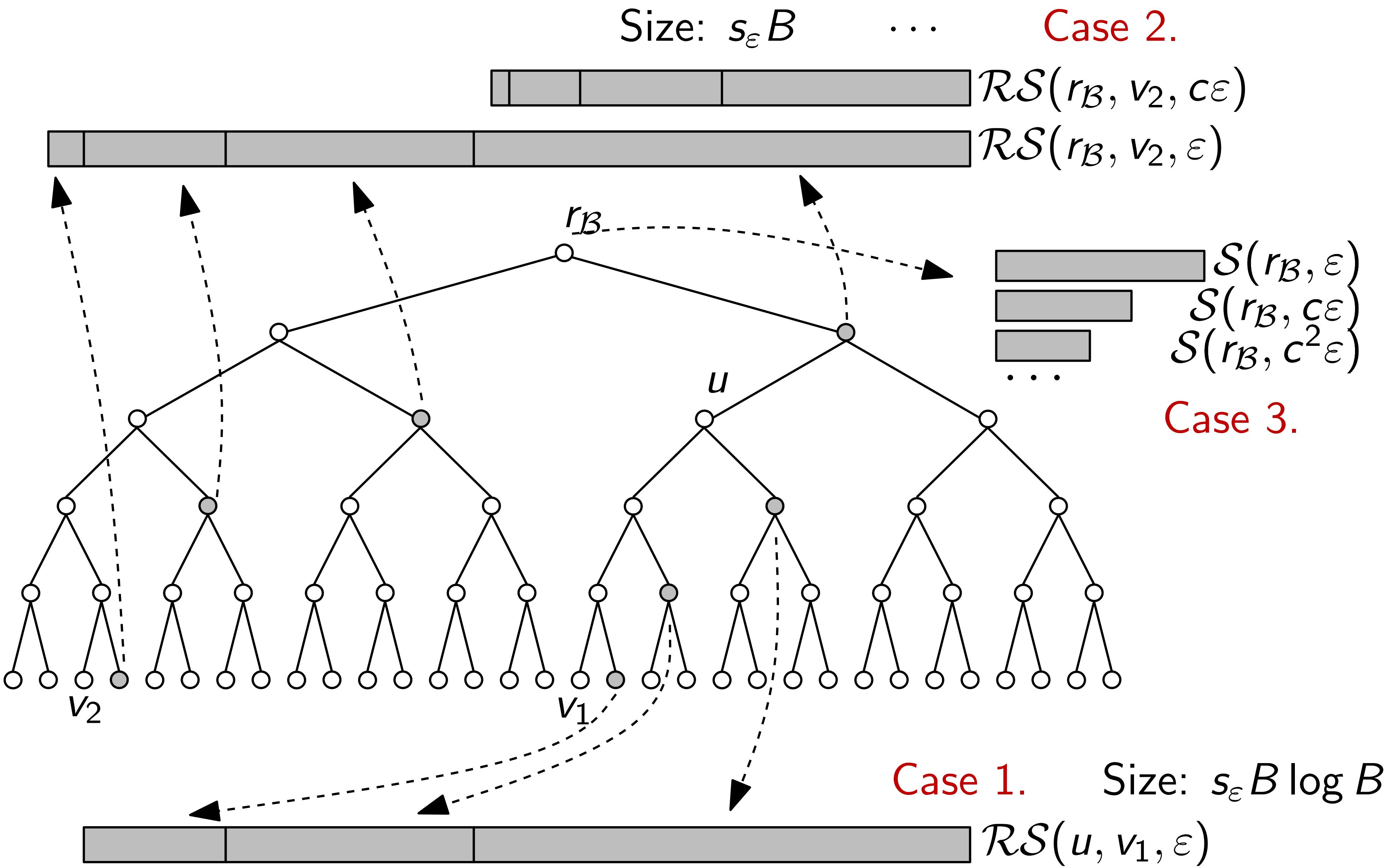
Focus on a Block



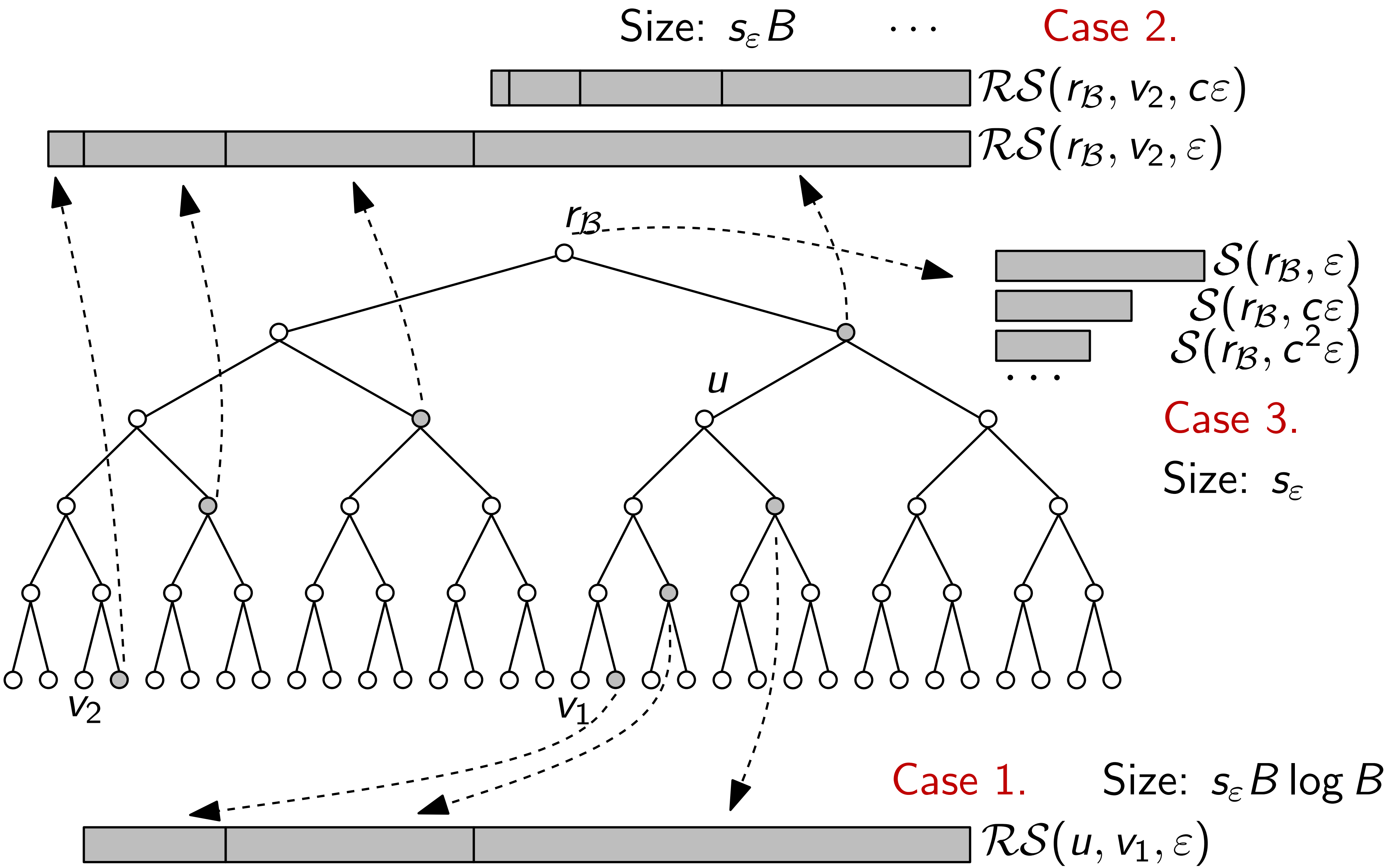
Focus on a Block



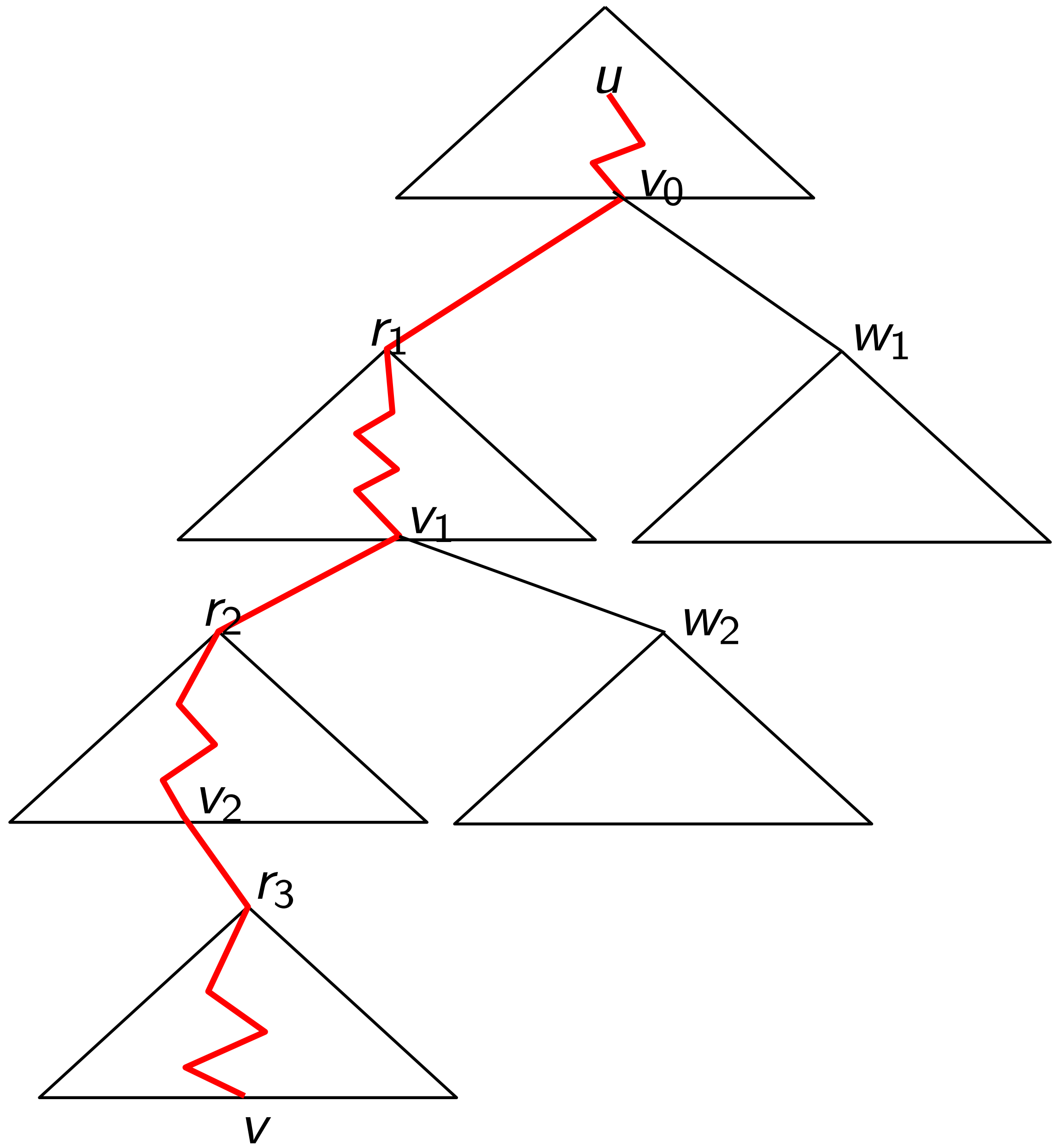
Focus on a Block



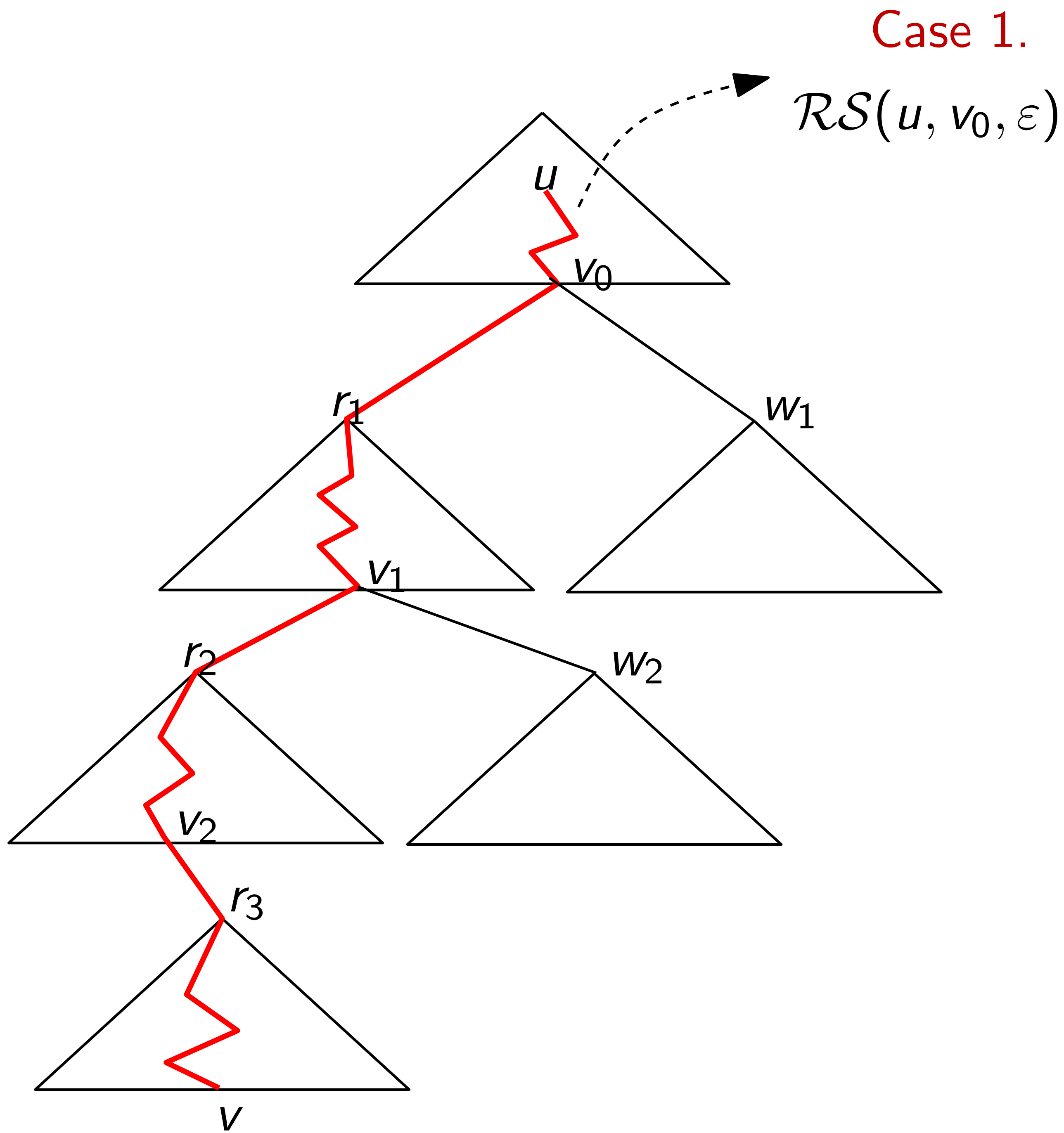
Focus on a Block



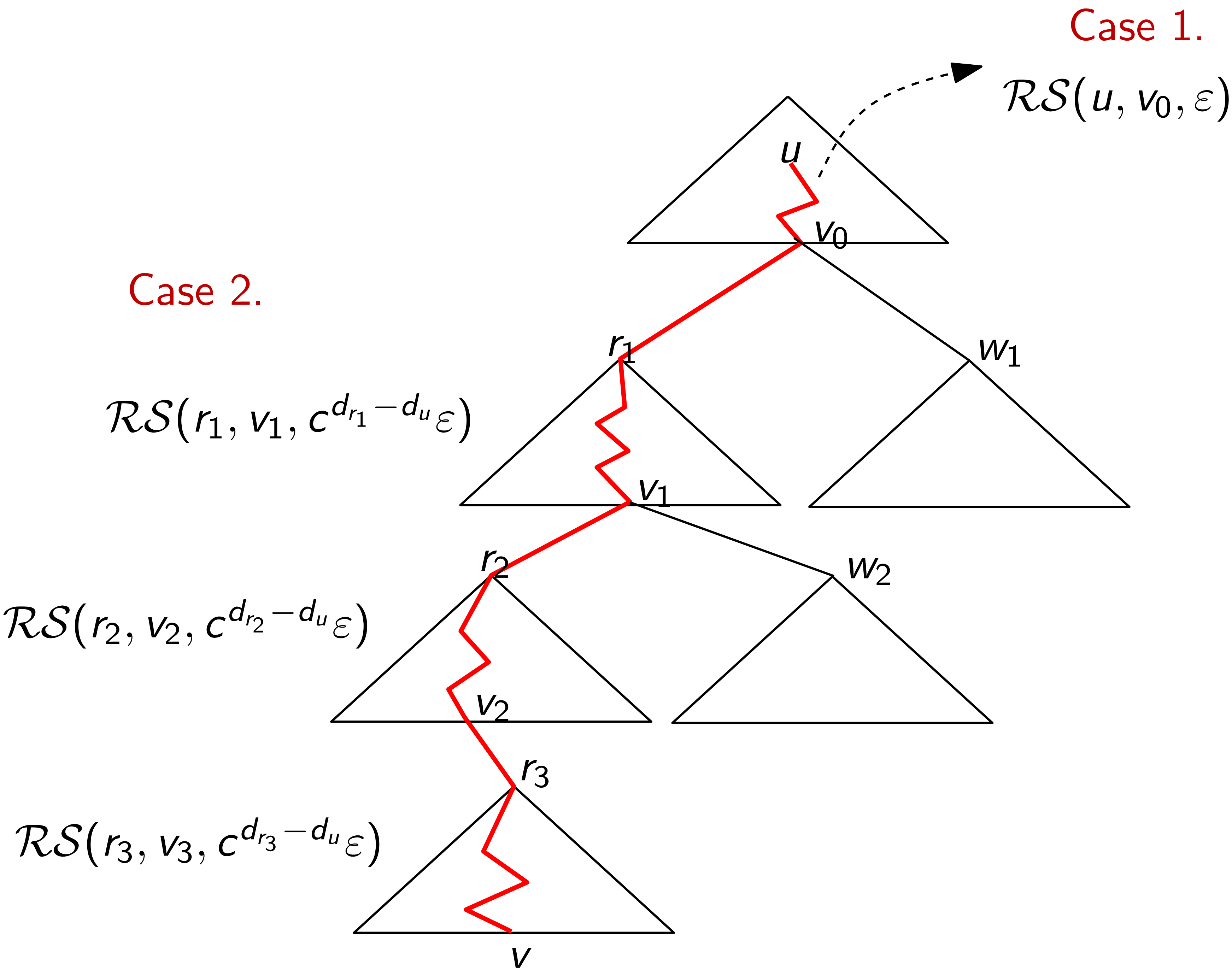
Query Process



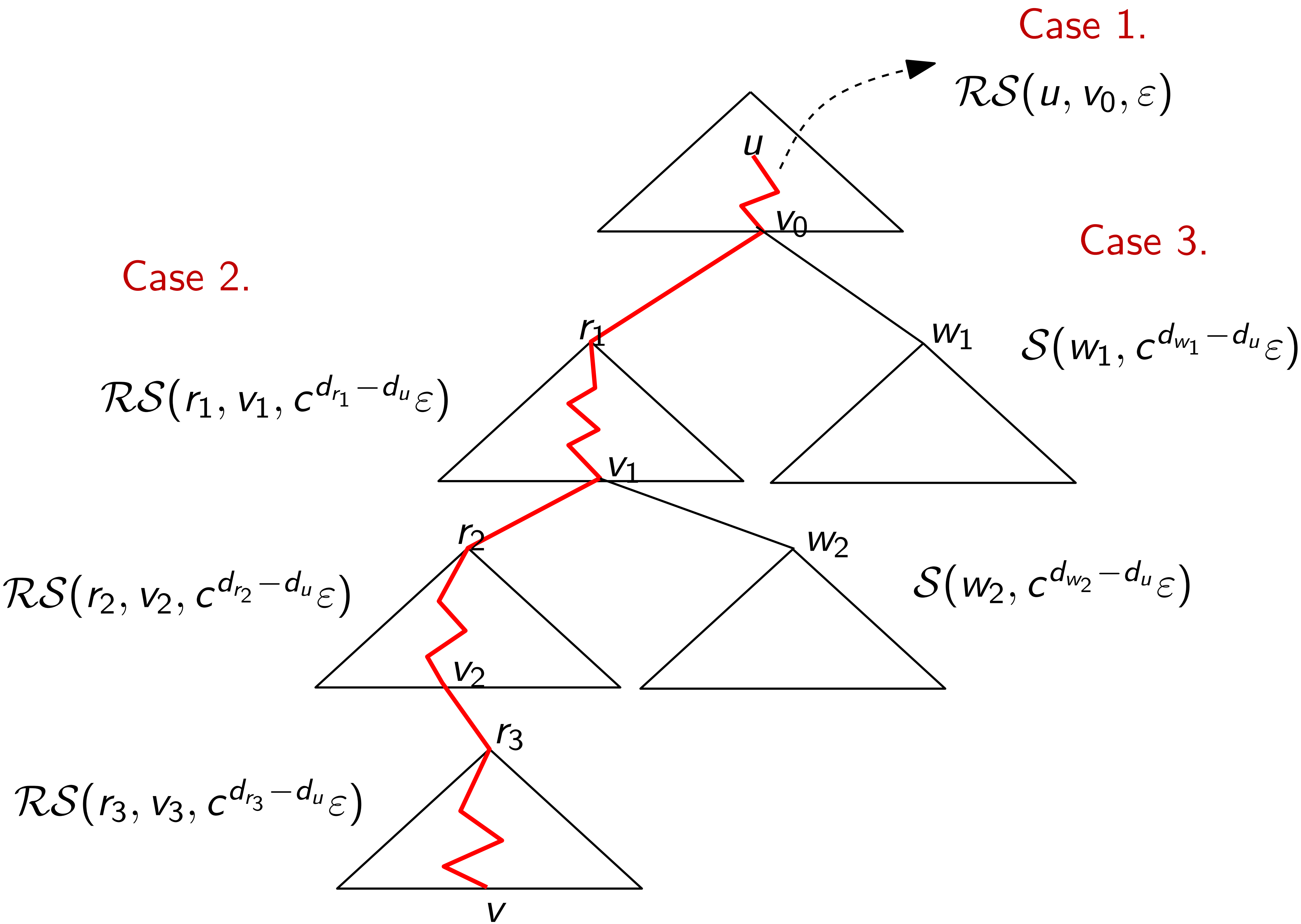
Query Process



Query Process

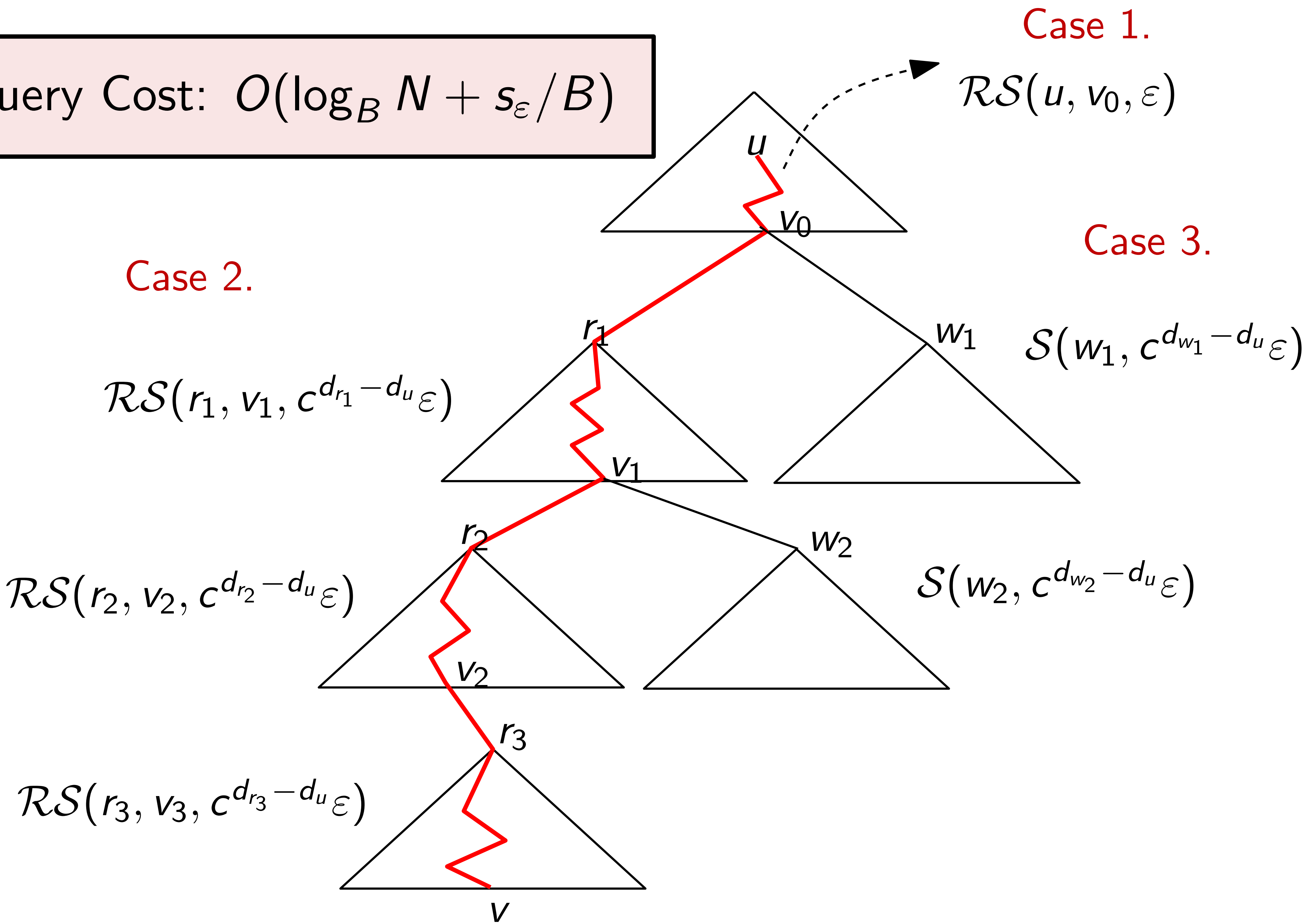


Query Process



Query Process

Query Cost: $O(\log_B N + s_\varepsilon / B)$



Optimal Data Structure - External Memory

Query Cost: $O(\log_B N + s_\varepsilon / B)$
Space Usage: $O(N \log B)$

Optimal Data Structure - External Memory

Query Cost: $O(\log_B N + s_\varepsilon / B)$
Space Usage: $O(N \log B)$

$\Downarrow$

Query Cost: $O(\log_B N + s_\varepsilon / B)$
Space Usage: $O(N)$

Optimal Data Structure - External Memory

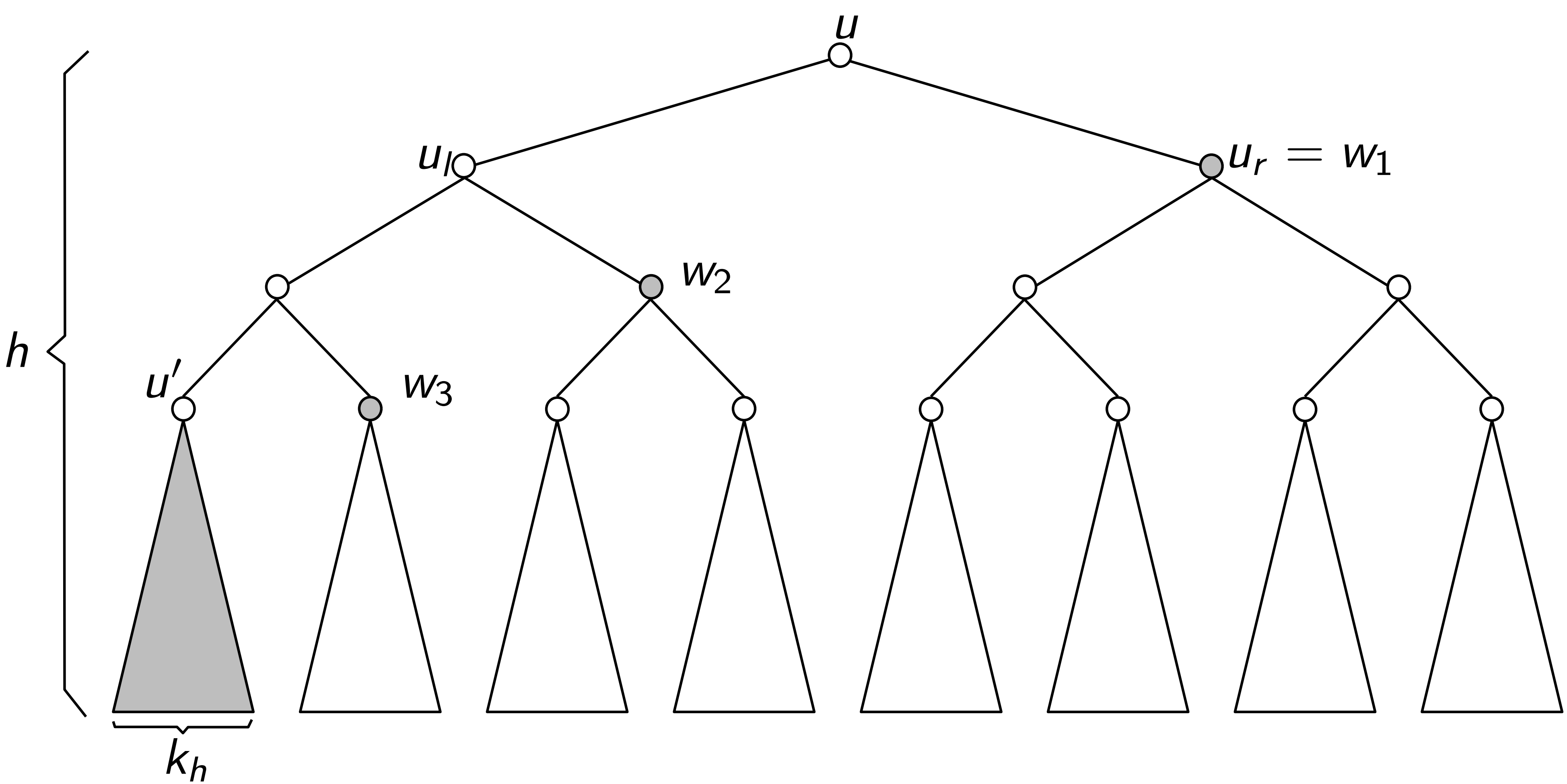
Query Cost: $O(\log_B N + s_\varepsilon / B)$
Space Usage: $O(N \log B)$

$\Downarrow$

Query Cost: $O(\log_B N + s_\varepsilon / B)$
Space Usage: $O(N)$

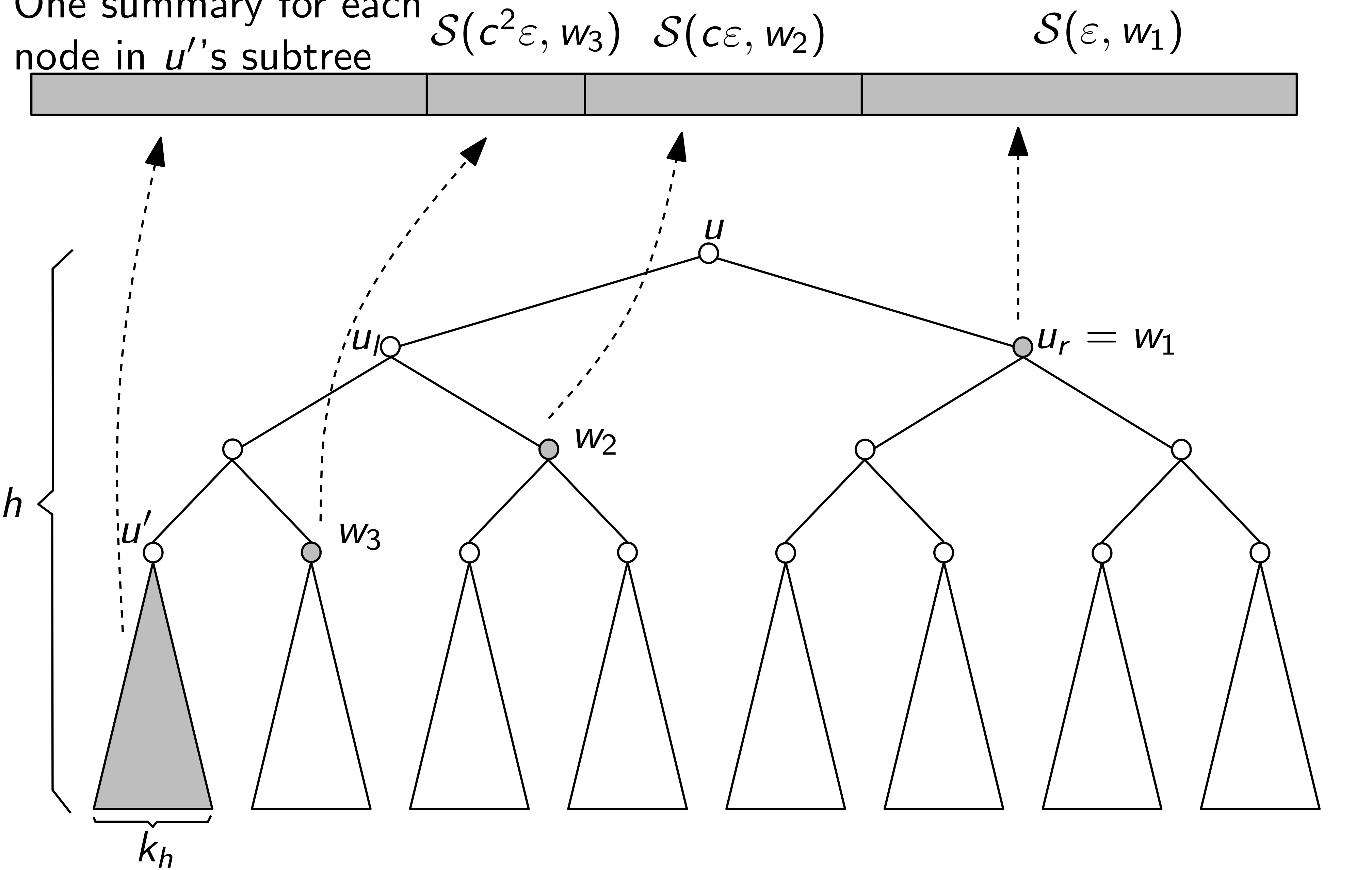
Idea: pack some leaves of u to reduce space usage

Packed Structure



Packed Structure

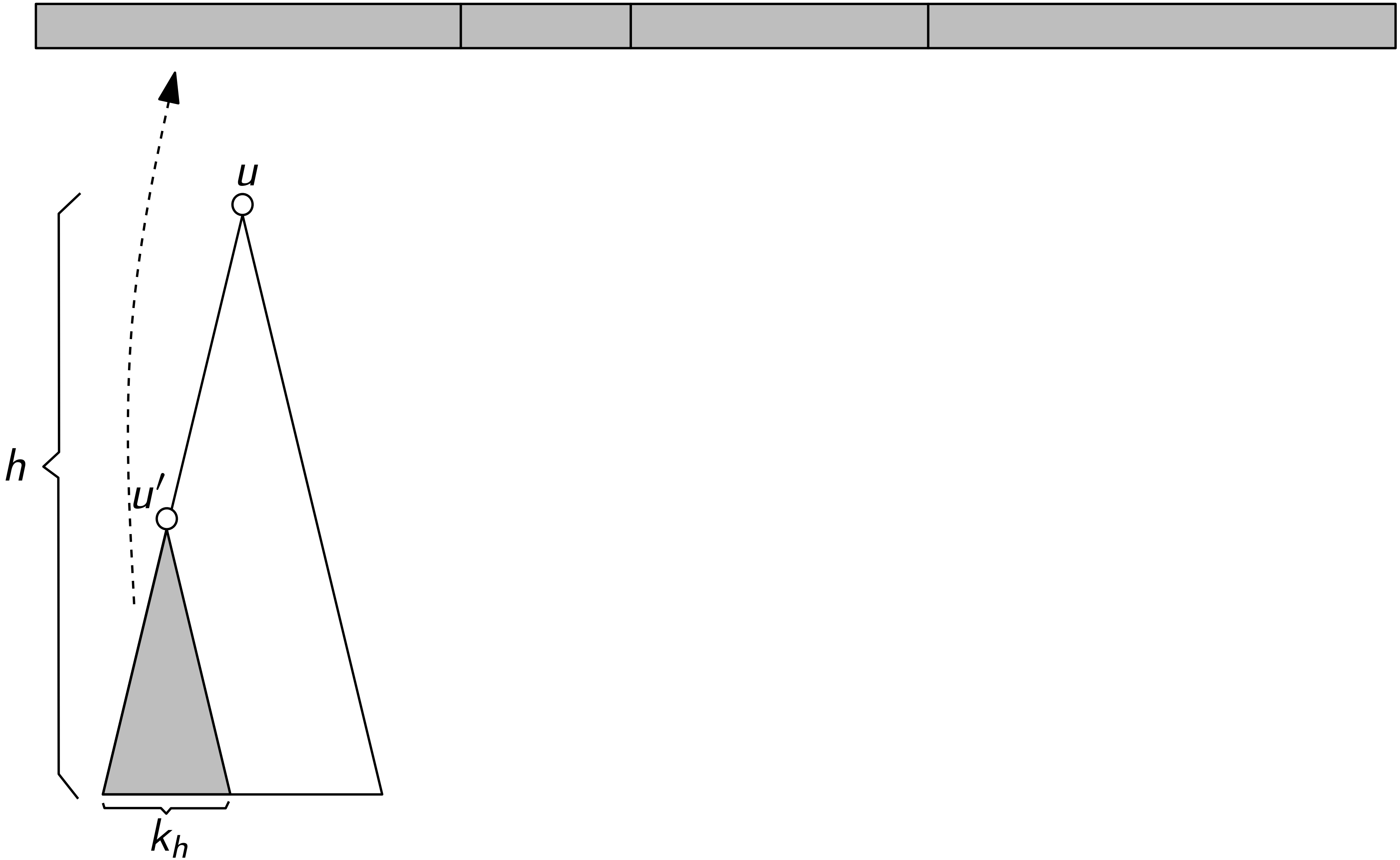
One summary for each node in u''s subtree



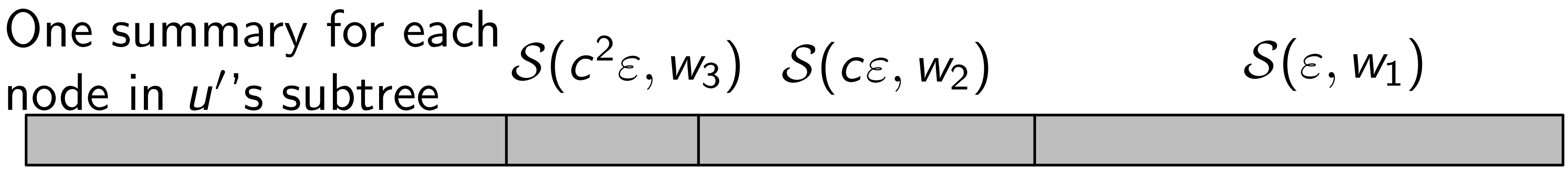
Packed Structure

One summary for each node in u''s subtree

$\mathcal{S}(c^2\varepsilon, w_3)$ $\mathcal{S}(c\varepsilon, w_2)$ $\mathcal{S}(\varepsilon, w_1)$

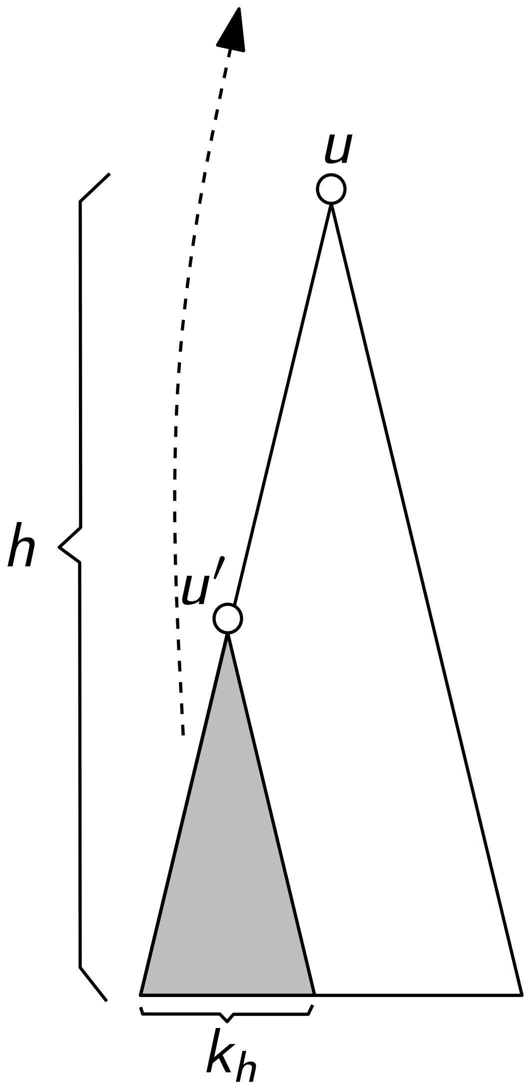


Packed Structure

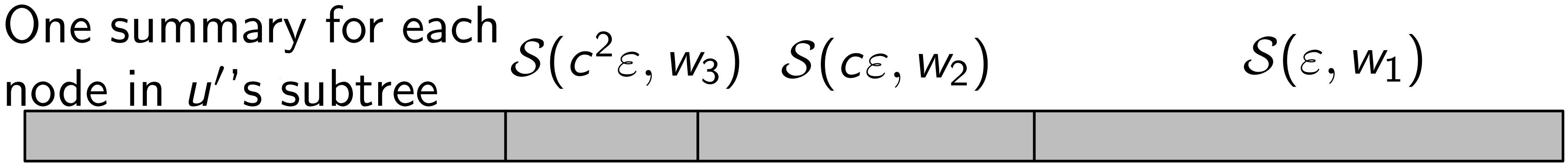


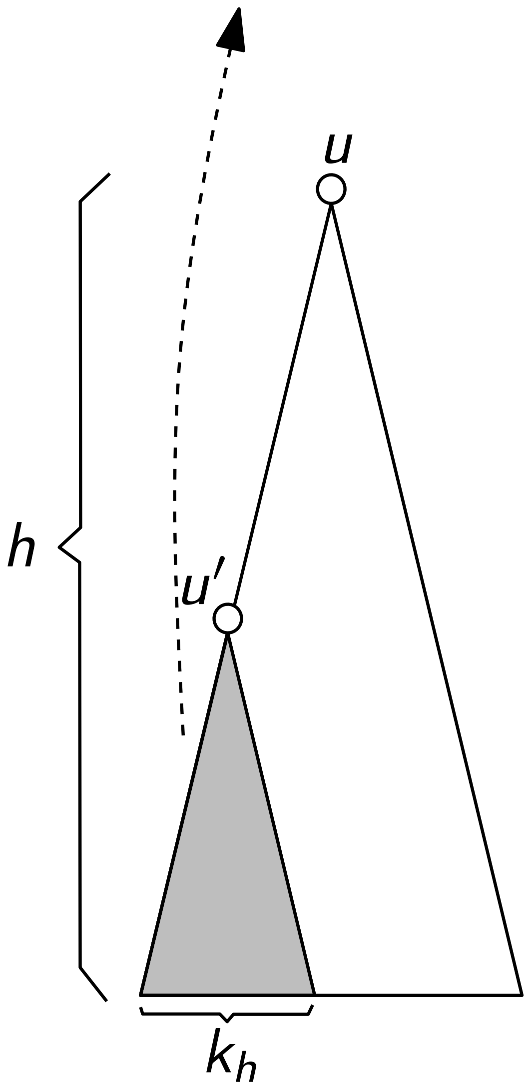
The total size of all summaries below u':

$$\sum_{i=0}^{\log k_h} \frac{k_h}{2^i} s_{c^{h-i-1}\varepsilon}. \quad (1)$$



Packed Structure



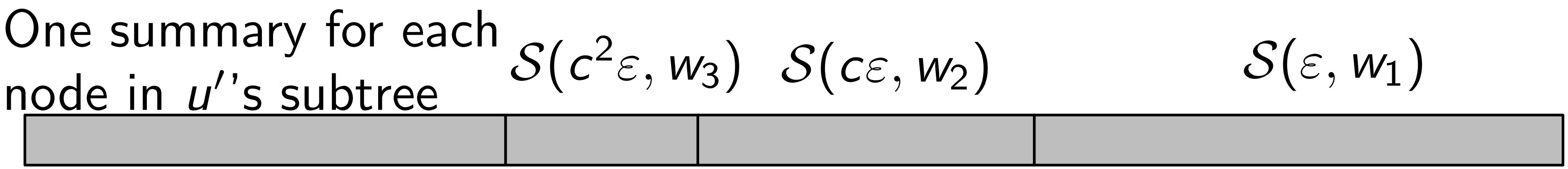


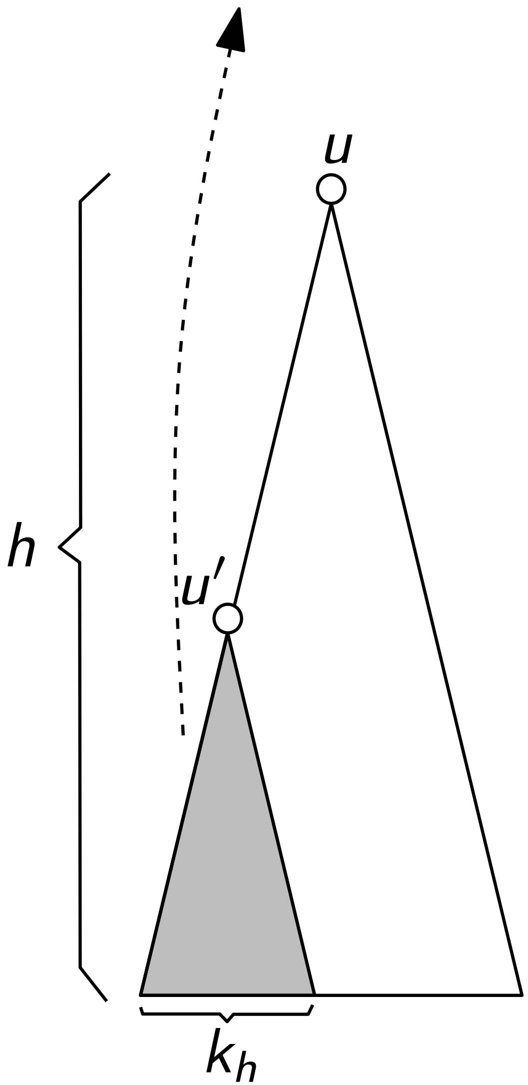
The total size of all summaries below u':

$$\sum_{i=0}^{\log k_h} \frac{k_h}{2^i} s_{c^{h-i-1}\varepsilon}. \qquad (1)$$

Choose k_h such that (1) is $\Theta(s_\varepsilon)$.

Packed Structure





The total size of all summaries below u':

$$\sum_{i=0}^{\log k_h} \frac{k_h}{2^i} s_{c^{h-i-1}\varepsilon}. \tag{1}$$

Choose k_h such that (1) is $\Theta(s_\varepsilon)$.

The total size of the packed structures in $\mathcal{B}$ is bounded by

$$\sum_{h=1}^{\log B} Bs_\varepsilon / k_h \leq O(Bs_\varepsilon).$$

Optimal Data Structure - External Memory

Theorem

For any (1/2)-exponentially decomposable summary, a database $\mathcal{D}$ of N records can be stored in an external memory index of linear size so that a summary query can be answered in $O(\log_B N + s_\varepsilon / B)$ I/Os.

Exponentially Decomposable vs. Decomposable

- Exponentially decomposable summaries
  - Heavy hitters
  - Quantile
  - Count-Min Sketch

Exponentially Decomposable vs. Decomposable

- Exponentially decomposable summaries
  - Heavy hitters
  - Quantile
  - Count-Min Sketch

Internal Memory:
Query cost: $O(\log N + s_\varepsilon)$
Space: $O(N)$

External Memory:
Query cost: $O(\log_B N + s_\varepsilon / B)$
Space: $O(N)$

Exponentially Decomposable vs. Decomposable

- Decomposable
  - AMS Sketch
  - Wavelets

Exponentially Decomposable vs. Decomposable

- Decomposable
  - AMS Sketch
  - Wavelets

Internal Memory:
Query cost: $O(s_\varepsilon \log N)$
Space: $O(N)$

External Memory:
Query cost: $O(\frac{s_\varepsilon}{B} \log N)$ for $s_\varepsilon \geq B$
$O(\log N / \log(B/s_\varepsilon))$ for $s_\varepsilon < B$
Space: $O(N)$

Exponentially Decomposable vs. Decomposable

- Decomposable
  - AMS Sketch
  - Wavelets

Internal Memory:

Query cost: $O(s_\varepsilon \log N)$

Space: $O(N)$

Can we improve?

External Memory:

Query cost: $O(\frac{s_\varepsilon}{B} \log N)$ for $s_\varepsilon \geq B$

$O(\log N / \log(B/s_\varepsilon))$ for $s_\varepsilon < B$

Space: $O(N)$

Open Problems

- Are the structures practical?

Open Problems

- Are the structures practical?
- Multiple query attributes:
  - (Q4) Return a summary on the household income distribution for the area within 50 miles from Washington, DC.

Open Problems

- Are the structures practical?
- Multiple query attributes:
  - (Q4) Return a summary on the household income distribution for the area within 50 miles from Washington, DC.
- Multiple summary attributes:
  - (Q5) What is the geographical distribution of households with annual income below $50,000?
  - Geometric summaries: clustering, ε-approximations

Open Problems

- Are the structures practical?
- Multiple query attributes:
  - (Q4) Return a summary on the household income distribution for the area within 50 miles from Washington, DC.
- Multiple summary attributes:
  - (Q5) What is the geographical distribution of households with annual income below $50,000?
  - Geometric summaries: clustering, ε-approximations
- Joins? General SQL queries?

Thank you!