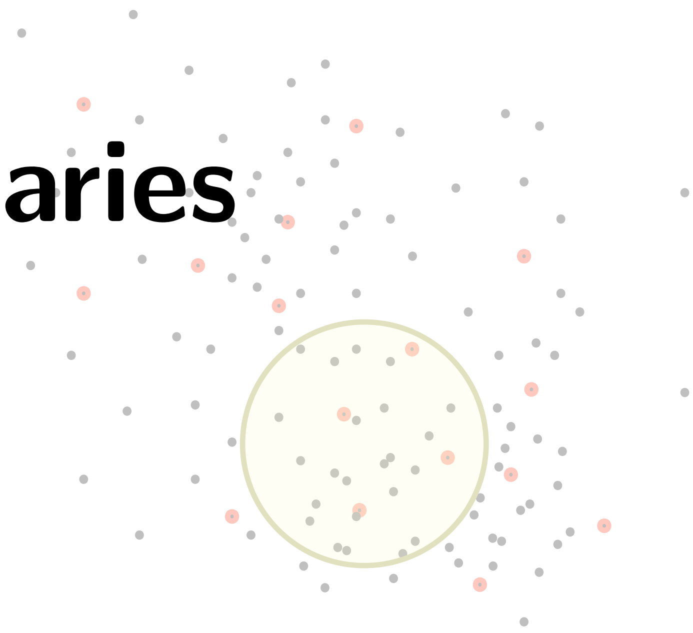
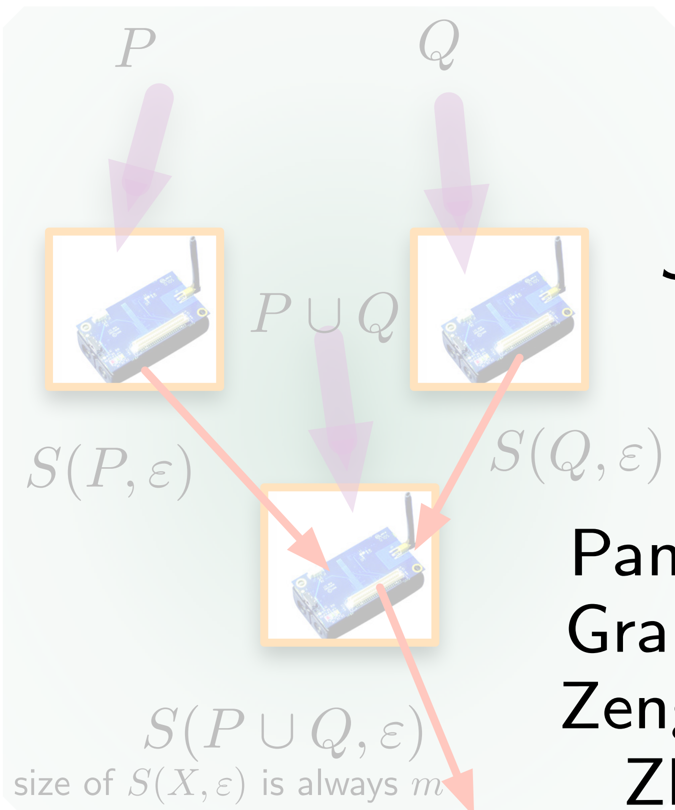
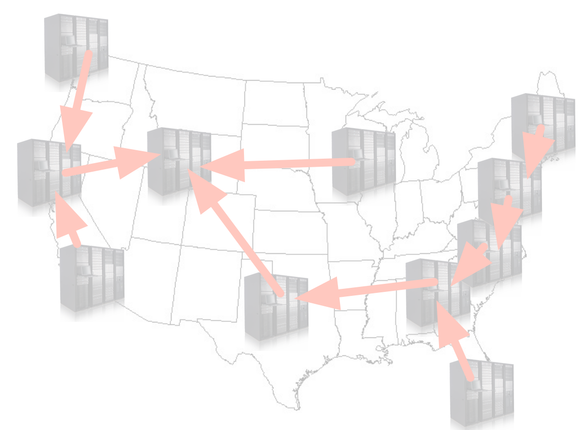
Mergeable Summaries

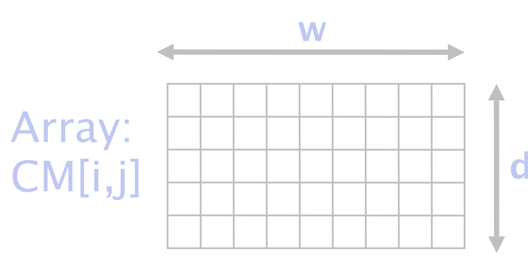




Jeff M. Phillips
University of Utah

joint with with
Pankaj K. Agarwal (Duke)
Graham Cormode (AT&T)
Zengfeng Huang (HKUST)
Zheiwei Wei (HKUST)
Ke Yi (HKUST)



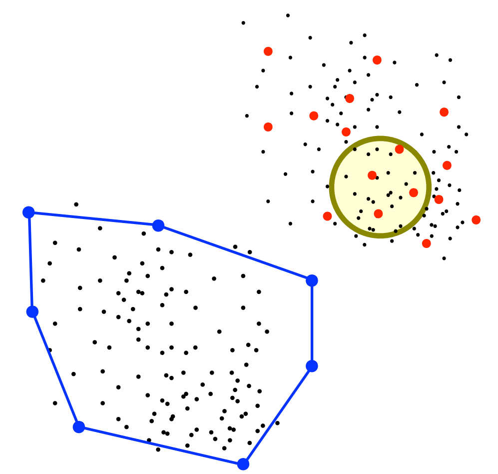


Summaries for MASSIVE Data

Allows approximate computation with guarantees and small space

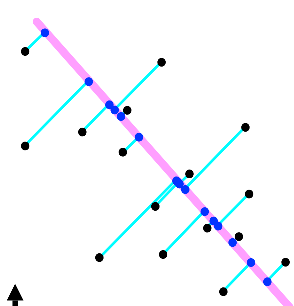
coreset: small summary, proxy for full data set
with approx guarantees:

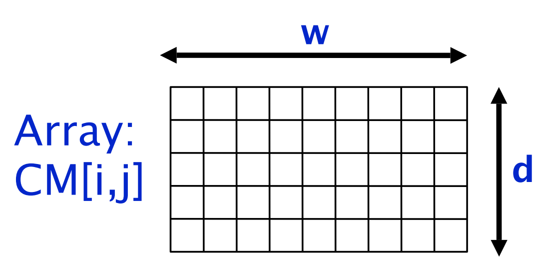
- ε-samples of $(P, \mathcal{R})$: approx density
- ε-kernel: approx convex shape



sketch: (random) (linear) combination of full data, recover functions
with approx guarantees:

- Euclidean distance: Johnson-Lindenstrauss random projection
- min-count sketch: approx item counts
- Greenwald-Khanna sketch: approx quantiles
- Misra-Gries sketch: approx frequent items





Summaries for MASSIVE Data

Allows approximate computation with guarantees and small space

Massive Distributed Computation

data centers

Massive Distributed Computation

data centers

Massive Distributed Computation

data centers

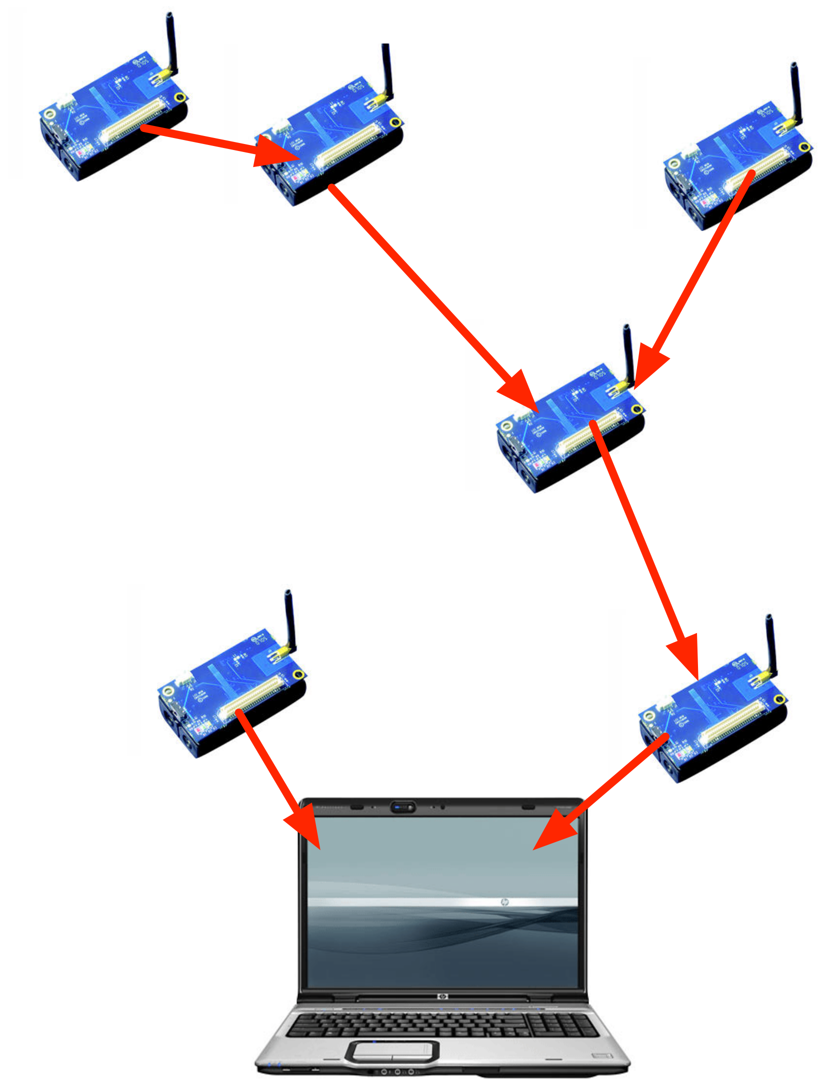
Massive Distributed Computation

data centers

Massive Distributed Computation

data centers

sensor networks

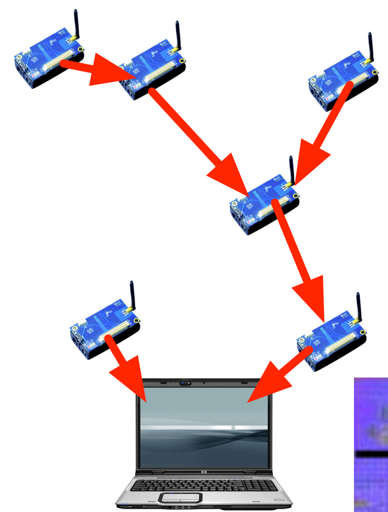


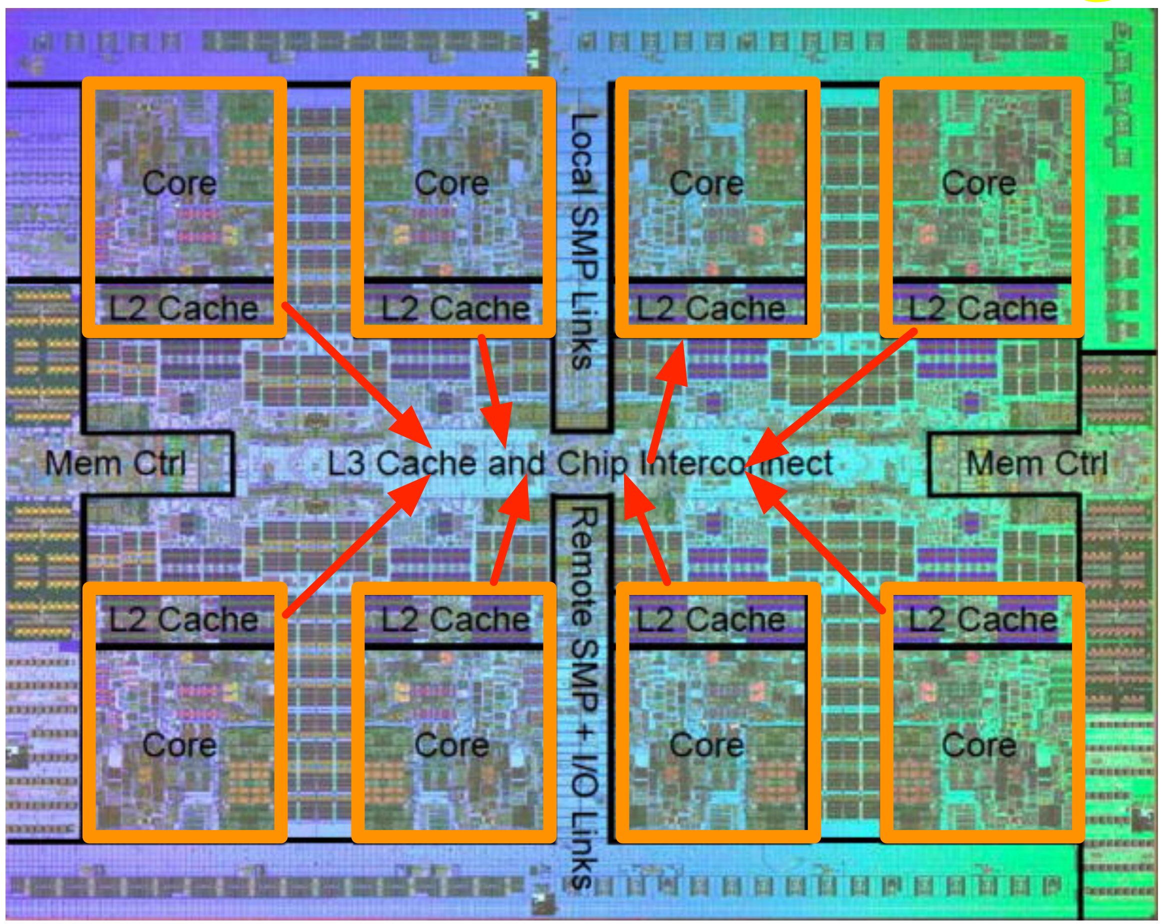
Massive Distributed Computation

data centers

sensor networks

multi-core



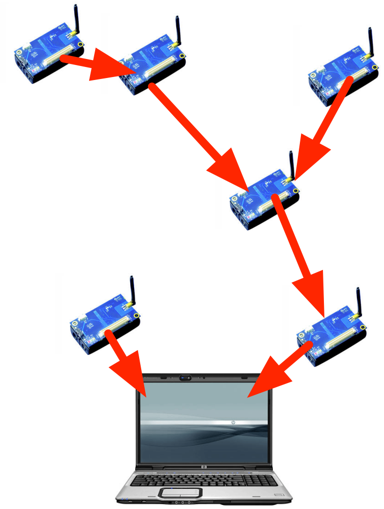


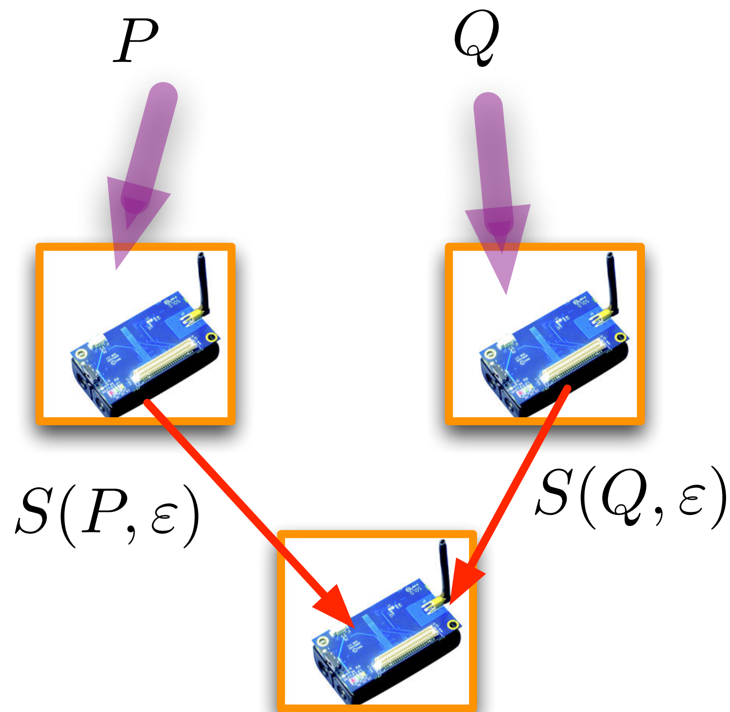
Massive Distributed Computation

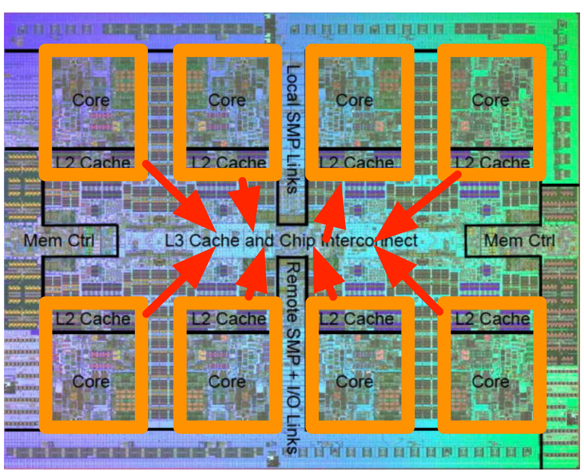
data centers

sensor networks

multi-core





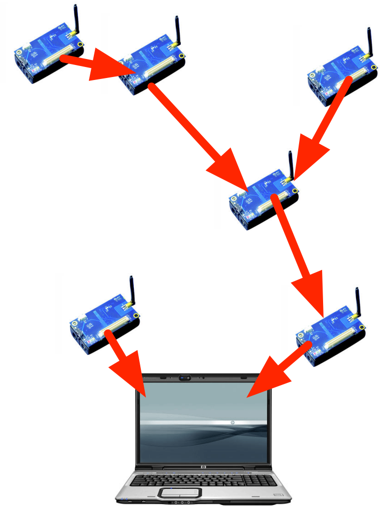


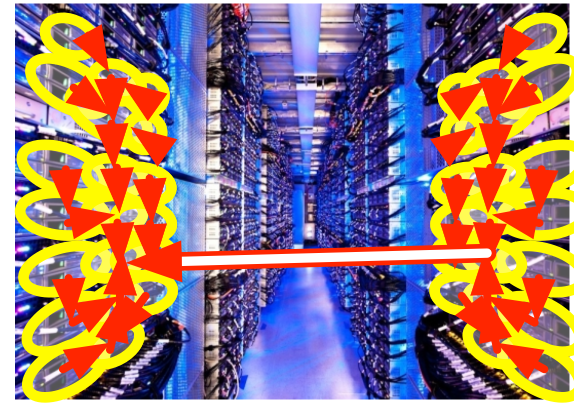
Massive Distributed Computation

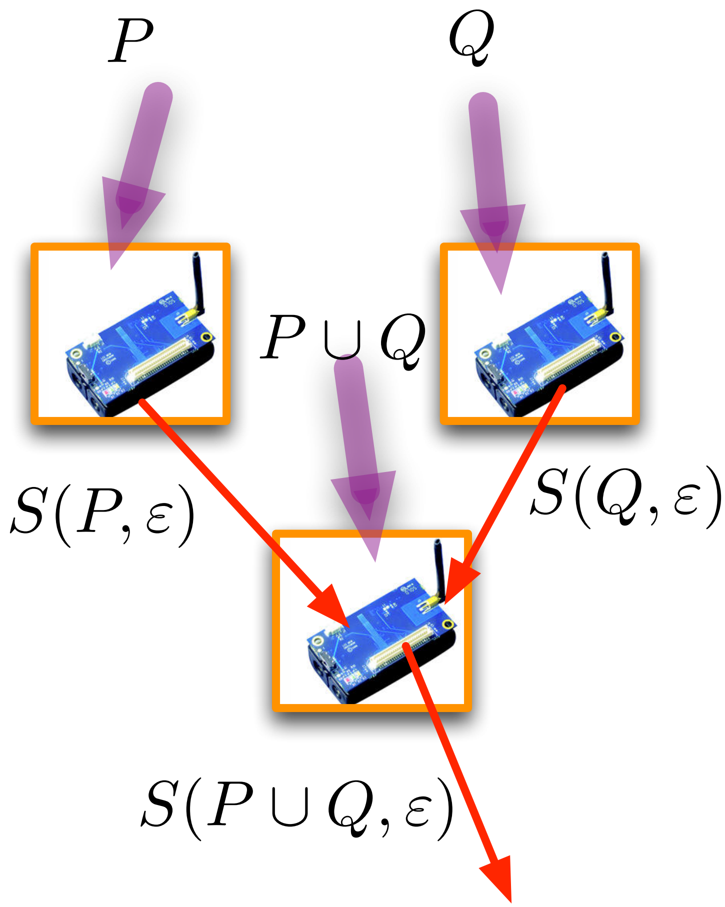
data centers

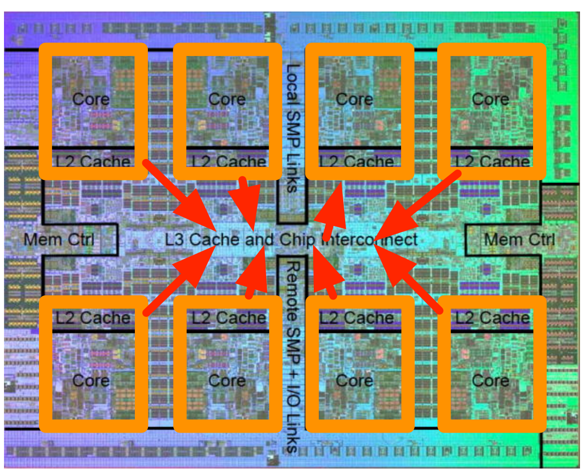
sensor networks

multi-core







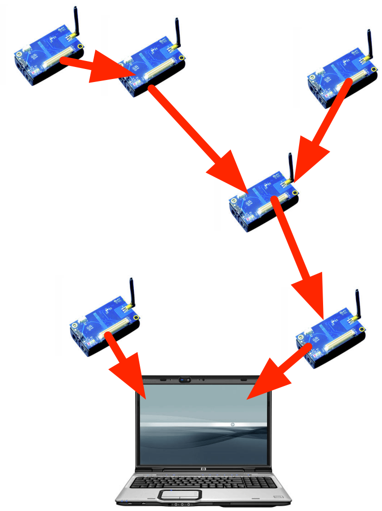


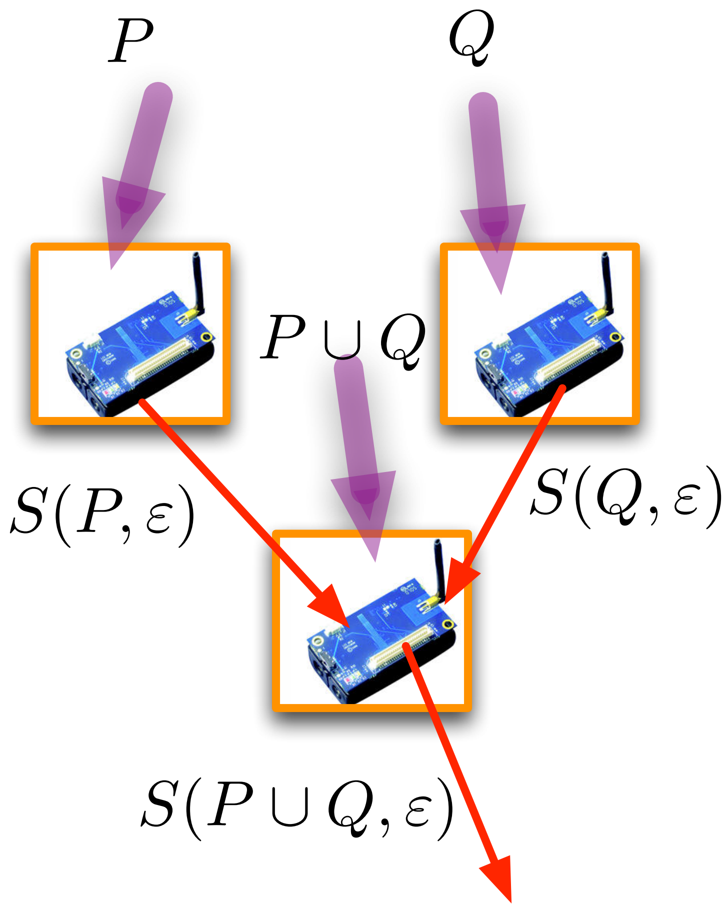
Massive Distributed Computation

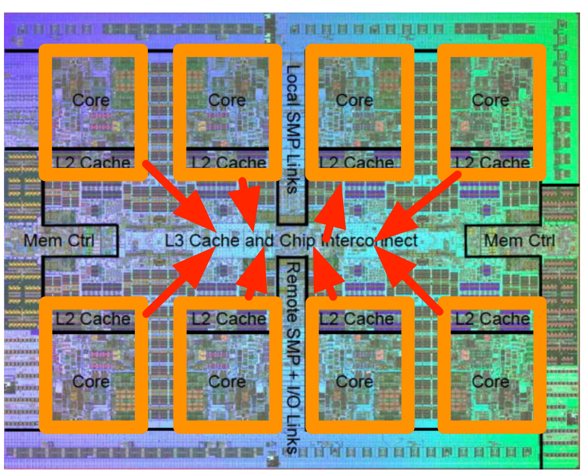
data centers

sensor networks

multi-core







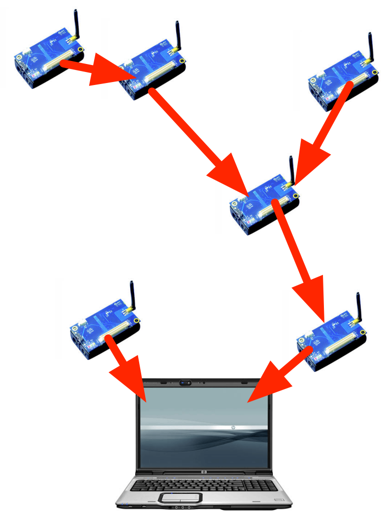
size of $S(X, \varepsilon)$ is always m

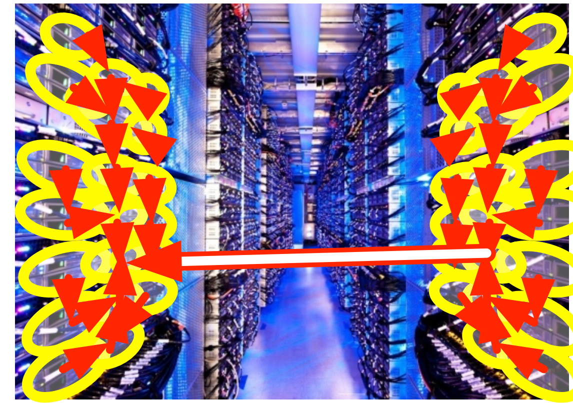
Massive Distributed Computation

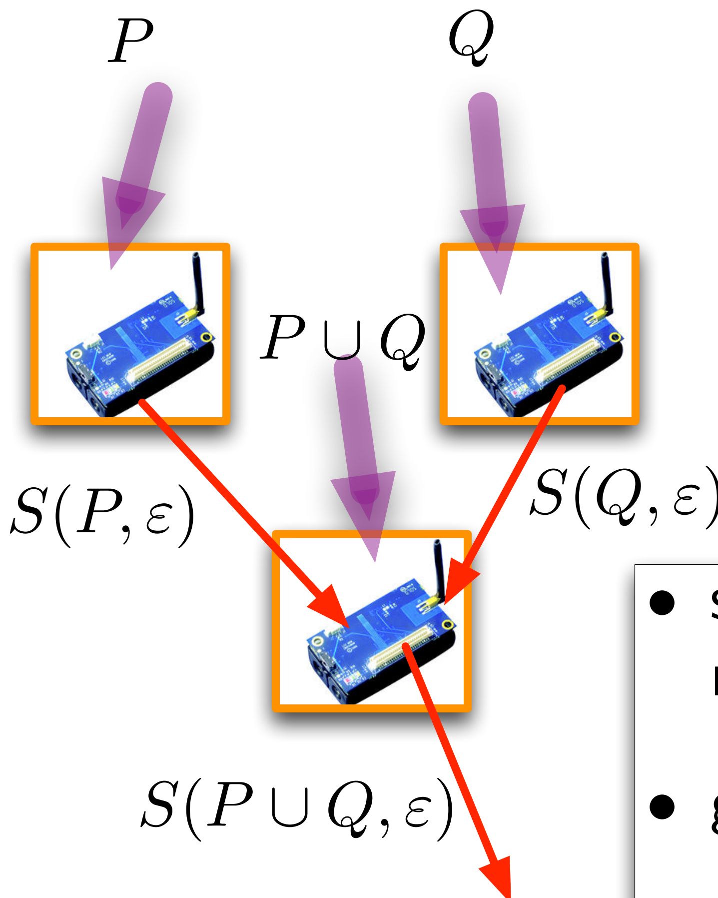
data centers

sensor networks

multi-core

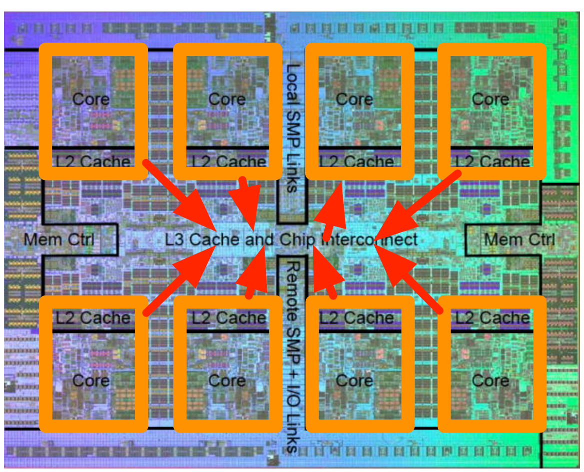




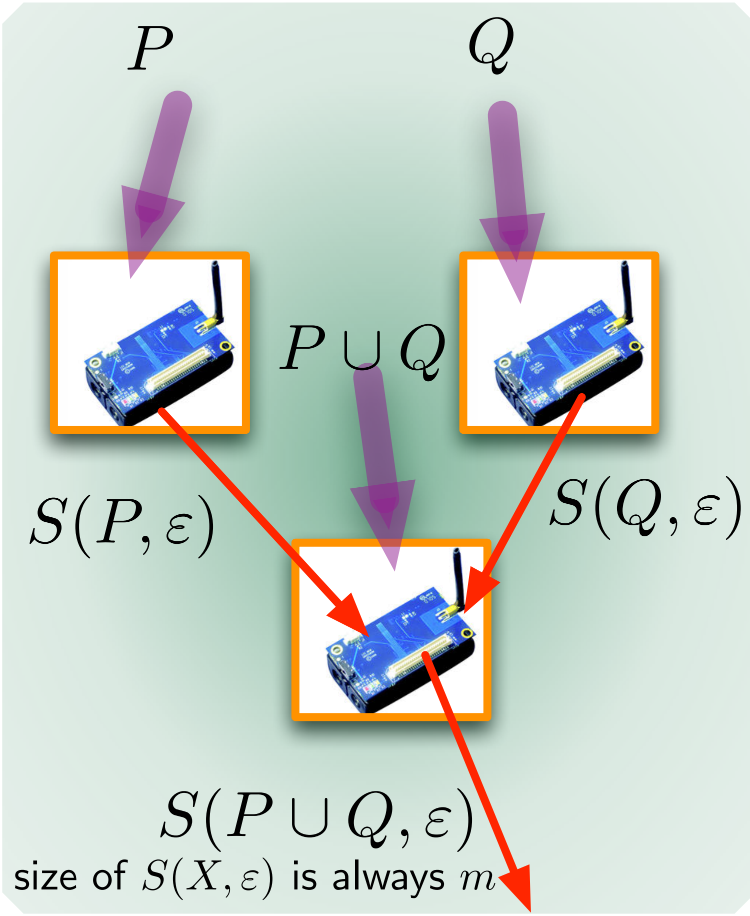


size of $S(X, \varepsilon)$ is always m

- similar to: MUD, Dremel
  more restrictive, "natural"
- generalizes streaming
- archiving summaries

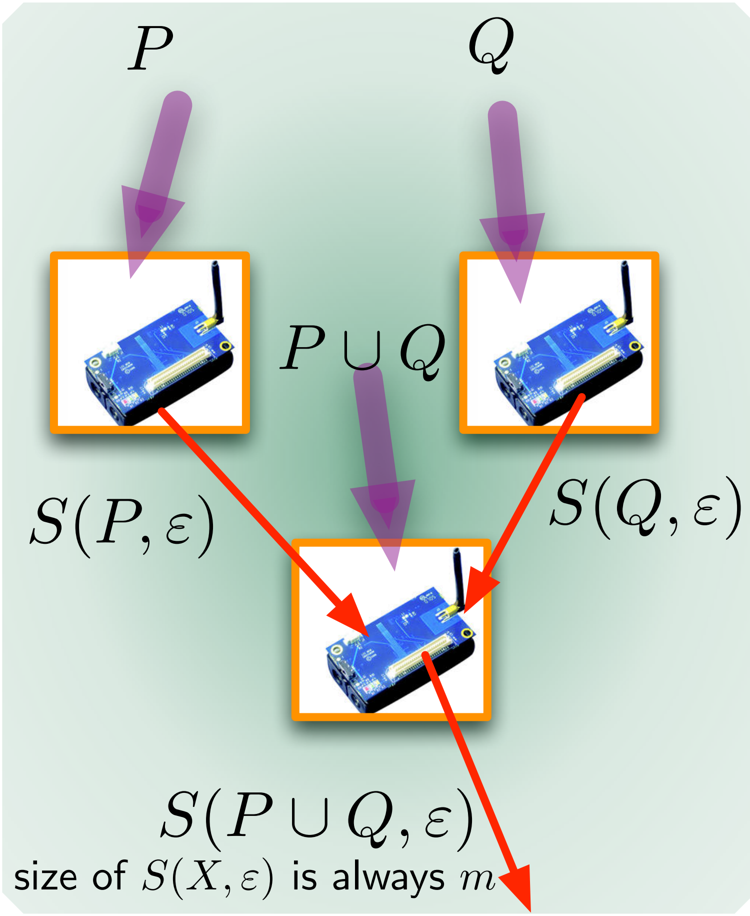


Random Sample



P	val	15	17	20	1	8	42	7	10	14	3

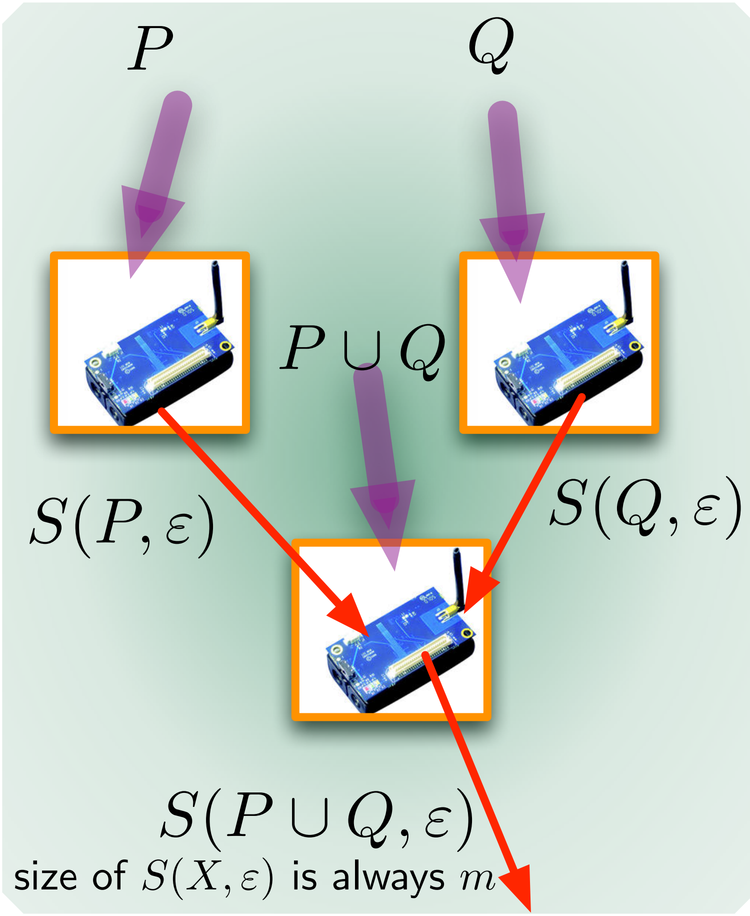
Random Sample



P

val	15	17	20	1	8	42	7	10	14	3
ran	.99	.42	.53	.01	.02	.23	.82	.75	.61	.14

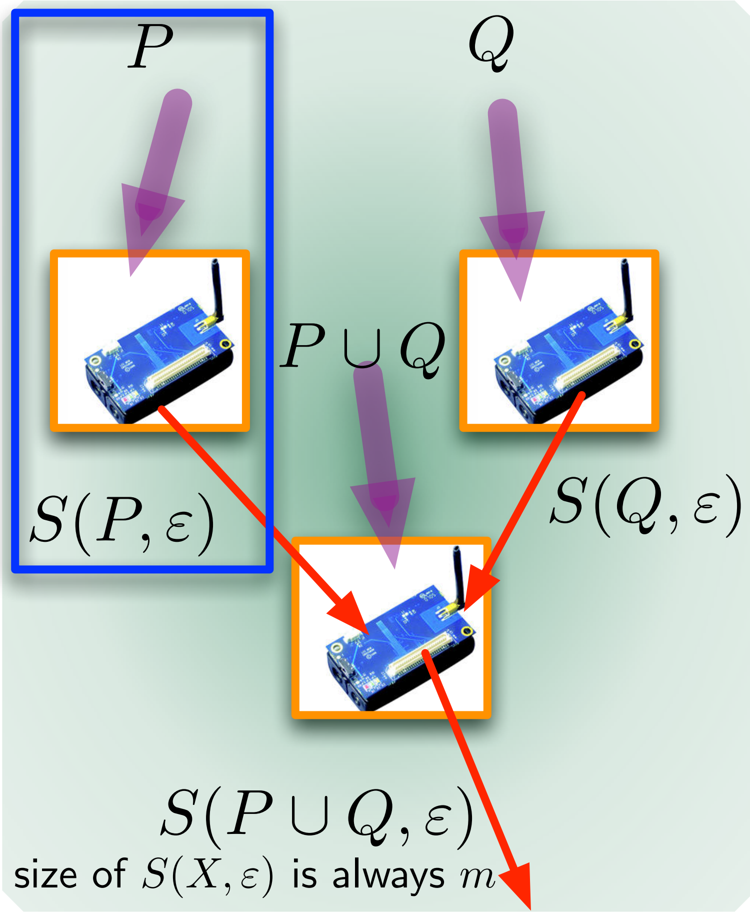
Random Sample



P

val	15	7	10	14	20	17	42	3	8	1
ran	.99	.82	.75	.61	.53	.42	.23	.14	.02	.01

Random Sample

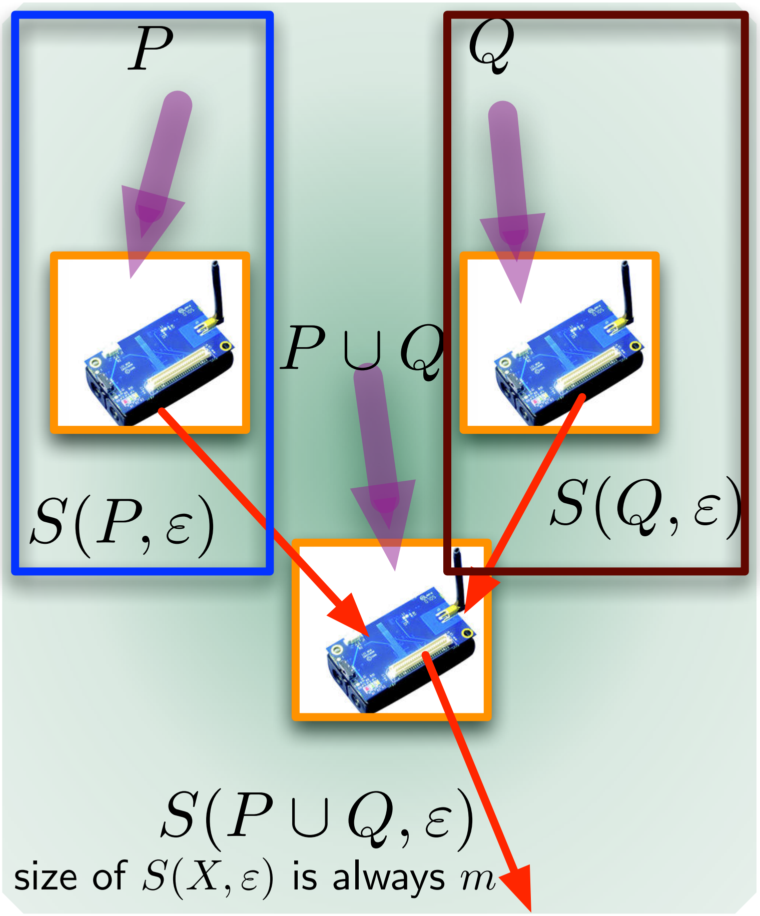


$S(P, \varepsilon)$

P

val	15	7	10	14	20	17	42	3	8	1
ran	.99	.82	.75	.61	.53	.42	.23	.14	.02	.01

Random Sample



$S(P, \varepsilon)$

P

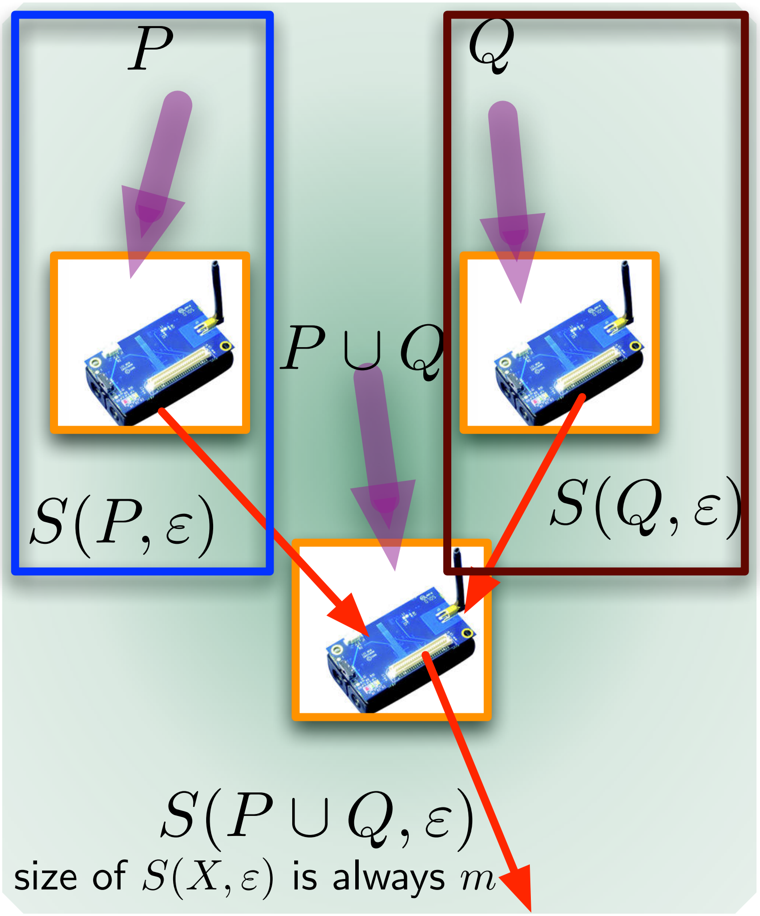
val	15	7	10	14	20	17	42	3	8	1
ran	.99	.82	.75	.61	.53	.42	.23	.14	.02	.01

$S(Q, \varepsilon)$

Q

val	31	9	16	11	14	7	2	13	21	4
ran	.90	.85	.80	.57	.50	.37	.31	.12	.10	.08

Random Sample



P — $S(P, \varepsilon)$ (first three value columns)

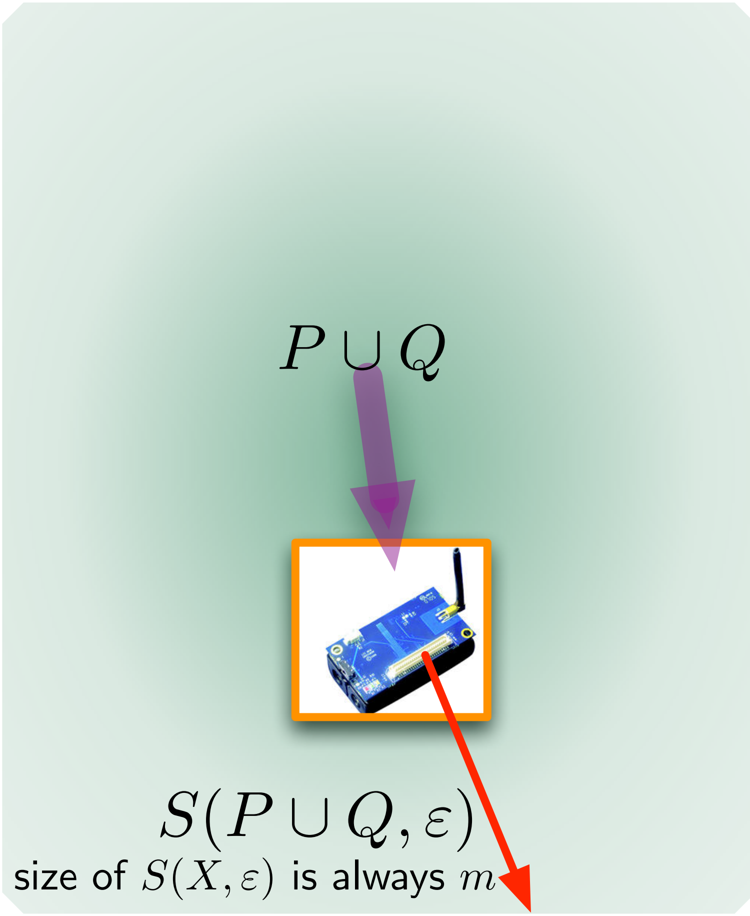
val	15	7	10	14	20	17	42	3	8	1
ran	.99	.82	.75	.61	.53	.42	.23	.14	.02	.01

Q — $S(Q, \varepsilon)$ (first three value columns)

val	31	9	16	11	14	7	2	13	21	4
ran	.90	.85	.80	.57	.50	.37	.31	.12	.10	.08

val	15	31	9	7	16	10
ran	.99	.90	.85	.82	.80	.75

Random Sample



$S(P, \varepsilon)$

P

val	15	7	10	14	20	17	42	3	8	1
ran	.99	.82	.75	.61	.53	.42	.23	.14	.02	.01

$S(Q, \varepsilon)$

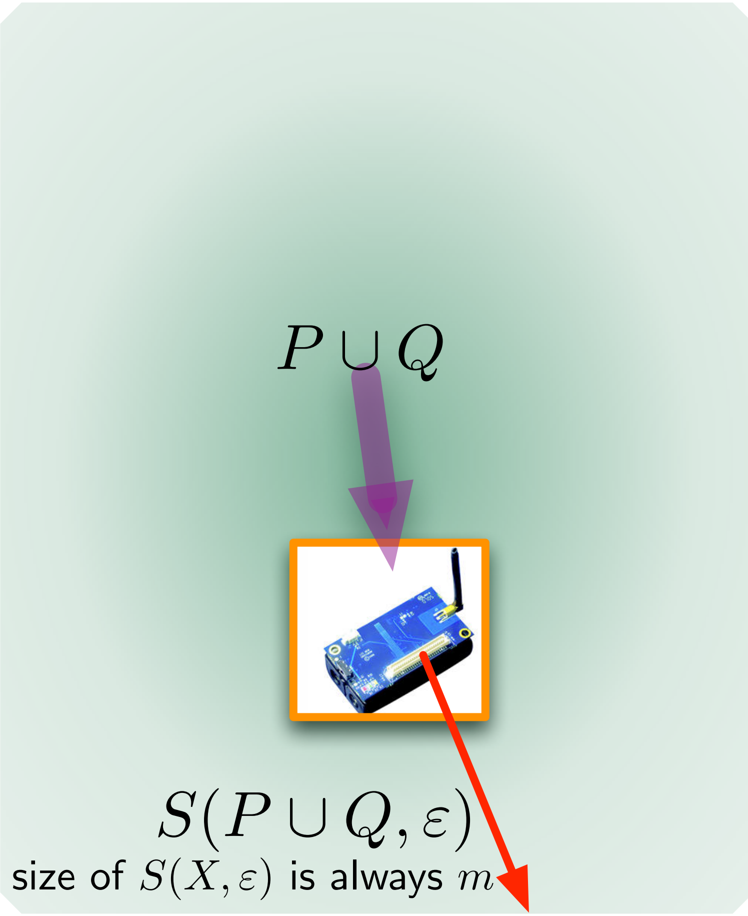
Q

val	31	9	16	11	14	7	2	13	21	4
ran	.90	.85	.80	.57	.50	.37	.31	.12	.10	.08

$S(P \cup Q, \varepsilon)$

val	15	31	9	7	16	10
ran	.99	.90	.85	.82	.80	.75

Random Sample



$S(P, \varepsilon)$

P

val	15	7	10	14	20	17	42	3	8	1
ran	.99	.82	.75	.61	.53	.42	.23	.14	.02	.01

$S(Q, \varepsilon)$

Q

val	31	9	16	11	14	7	2	13	21	4
ran	.90	.85	.80	.57	.50	.37	.31	.12	.10	.08

$S(P \cup Q, \varepsilon)$

val	15	31	9	7	16	10
ran	.99	.90	.85	.82	.80	.75

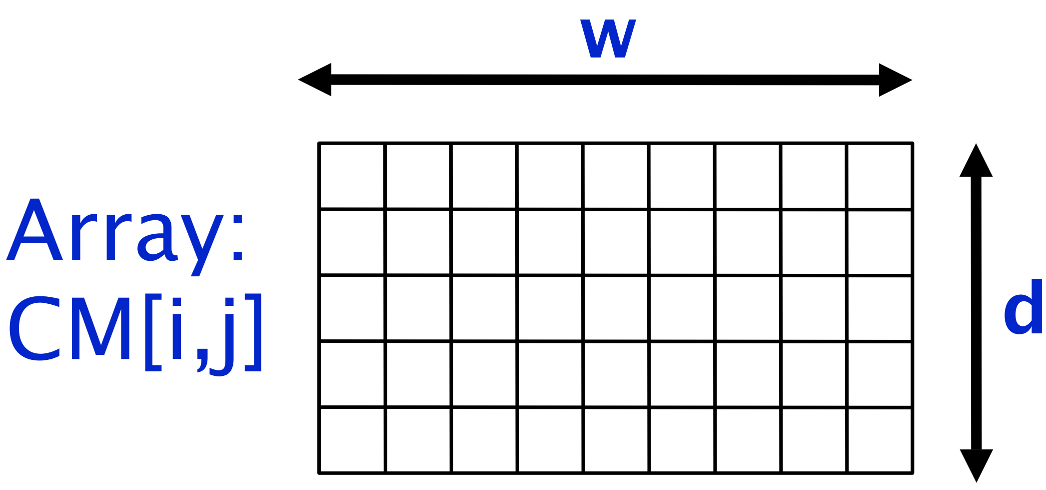
max element

top k elements

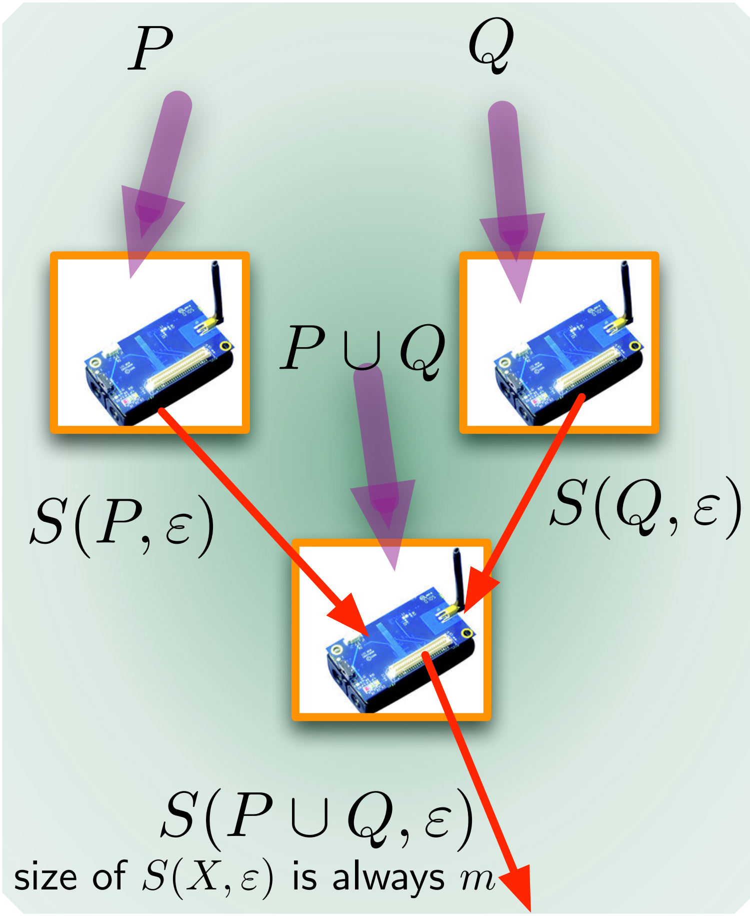
Linear Sketches

Count-Min sketch of vector $P[1...U]$:

- Linear sketch as array size $w \times d$
- Use d hash functions h to map x to $[1...w]$
- Estimate $P[i] = \min_j$ CM$[h_j(i), j]$



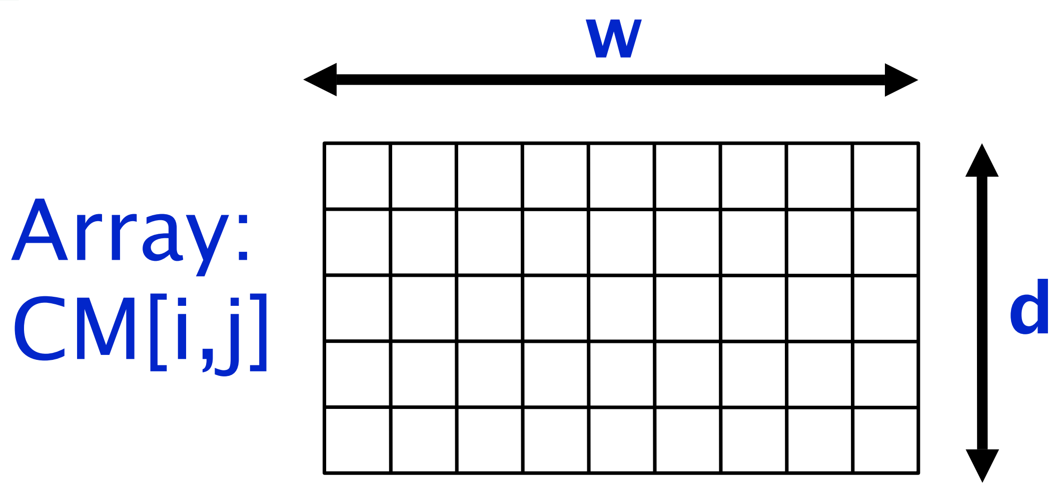
Linear Sketches



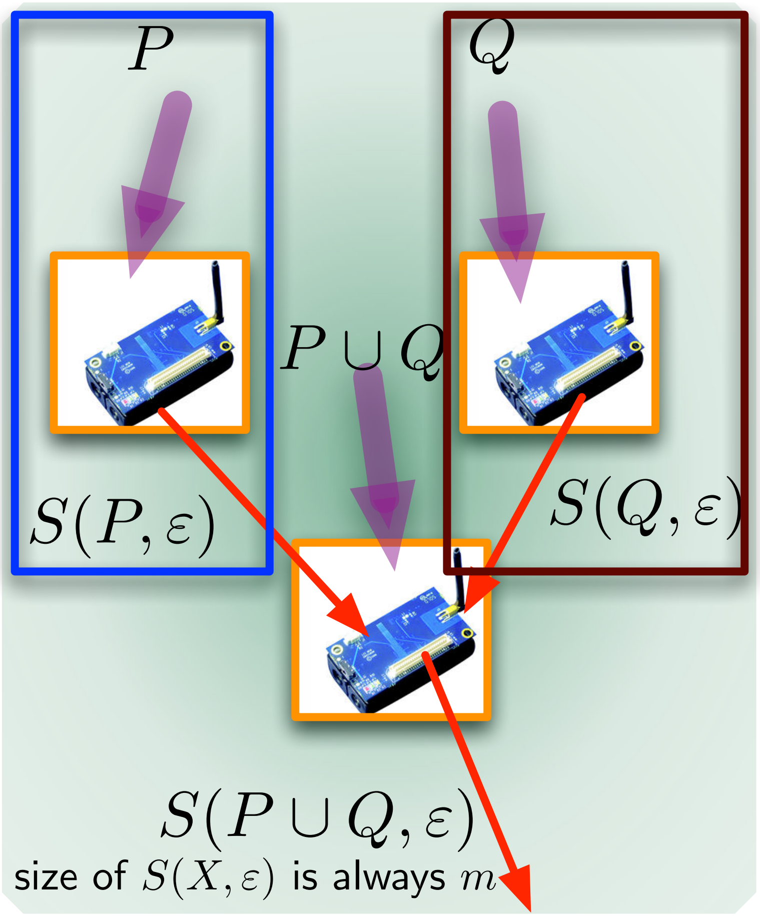
Count-Min sketch of vector $P[1 \ldots U]$:

- Linear sketch as array size $w \times d$
- Use d hash functions h to map x to $[1 \ldots w]$
- Estimate $P[i] = \min_j$ CM$[h_j(i), j]$

Mergeable: CM$(P + Q)$ = CM(P) + CM(Q)



Linear Sketches



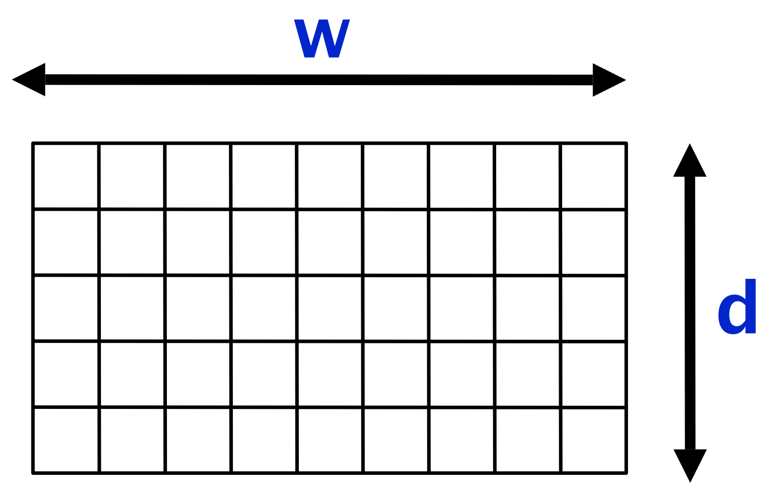
Count-Min sketch of vector $P[1 \ldots U]$:

- Linear sketch as array size $w \times d$
- Use d hash functions h to map x to $[1 \ldots w]$
- Estimate $P[i] = \min_j \text{CM}[h_j(i), j]$

Mergeable: $\text{CM}(P+Q) = \text{CM}(P) + \text{CM}(Q)$

$S(P, \varepsilon)$ $S(Q, \varepsilon)$

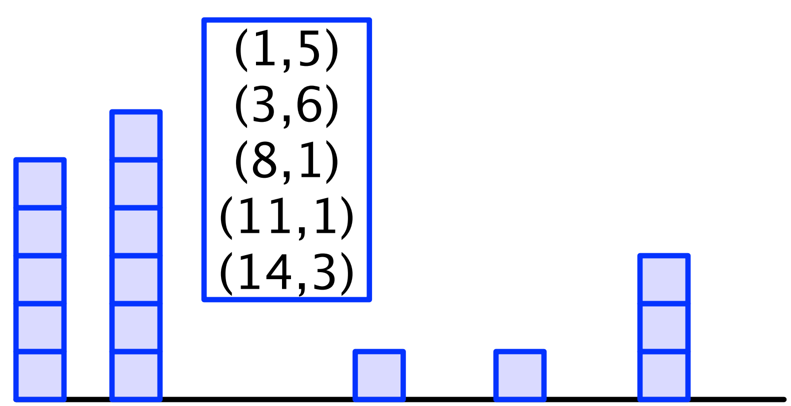
Array: CM[i,j]



Heavy Hitters Summaries

Misra-Gries (MG) sketch of $P[1...U]$:

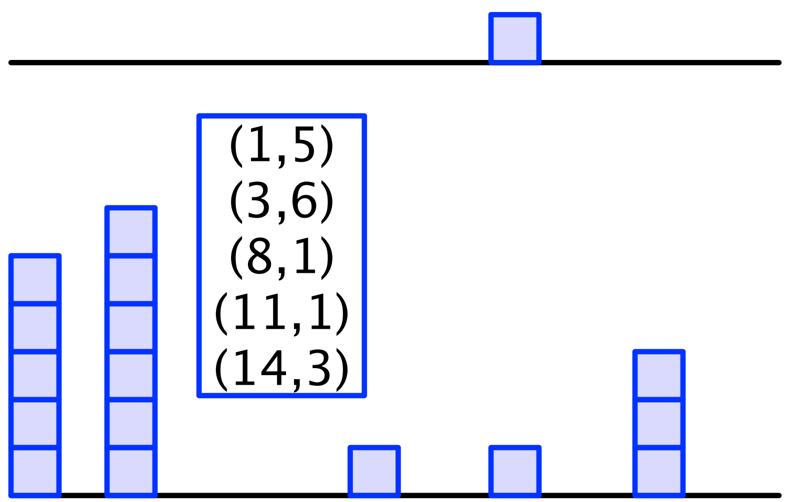
- Keep k (index,count) pairs
- If existing index arrives, update count
- If new index arrives, make new pair, or decrement all counts



Heavy Hitters Summaries

Misra-Gries (MG) sketch of $P[1...U]$:

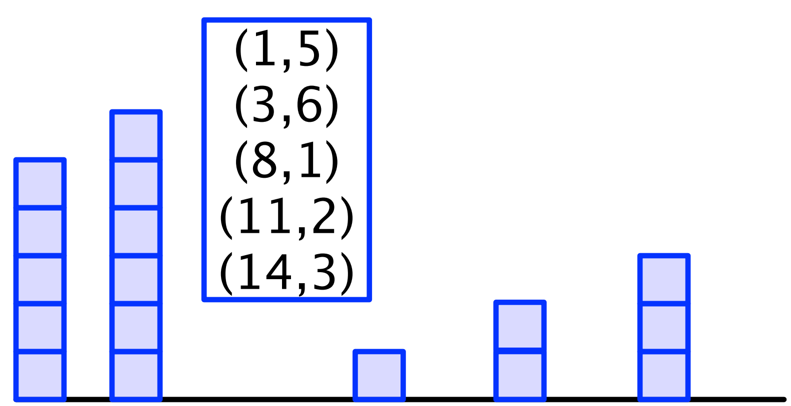
- Keep k (index,count) pairs
- If existing index arrives, update count
- If new index arrives, make new pair, or decrement all counts



Heavy Hitters Summaries

Misra-Gries (MG) sketch of $P[1...U]$:

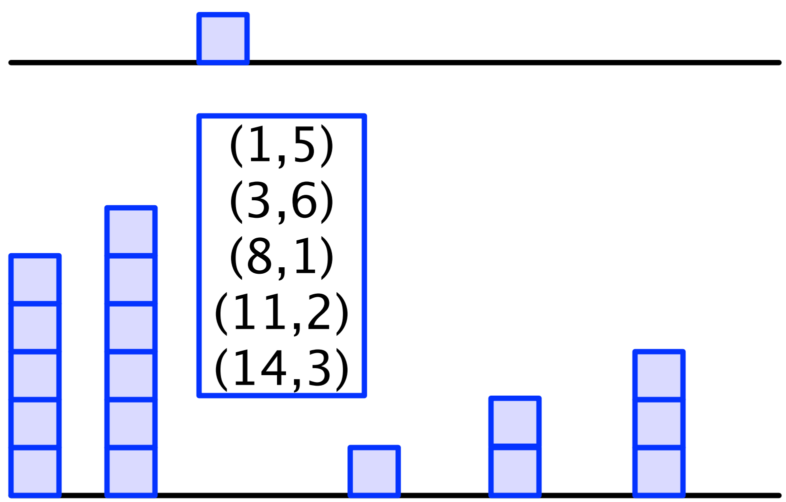
- Keep k (index,count) pairs
- If existing index arrives, update count
- If new index arrives, make new pair,
  or decrement all counts



Heavy Hitters Summaries

Misra-Gries (MG) sketch of $P[1...U]$:

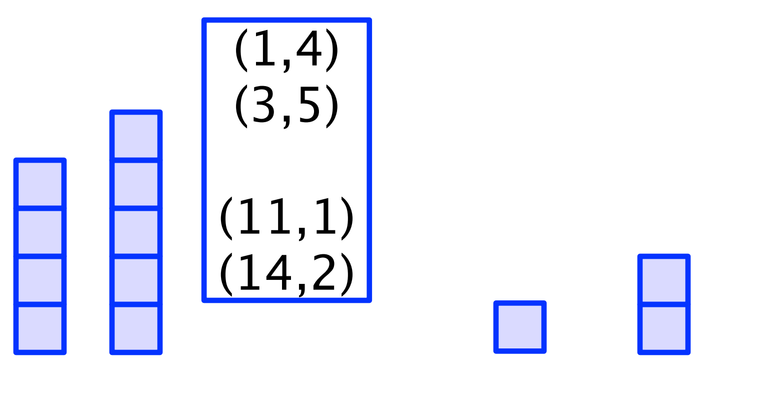
- Keep k (index,count) pairs
- If existing index arrives, update count
- If new index arrives, make new pair, or decrement all counts



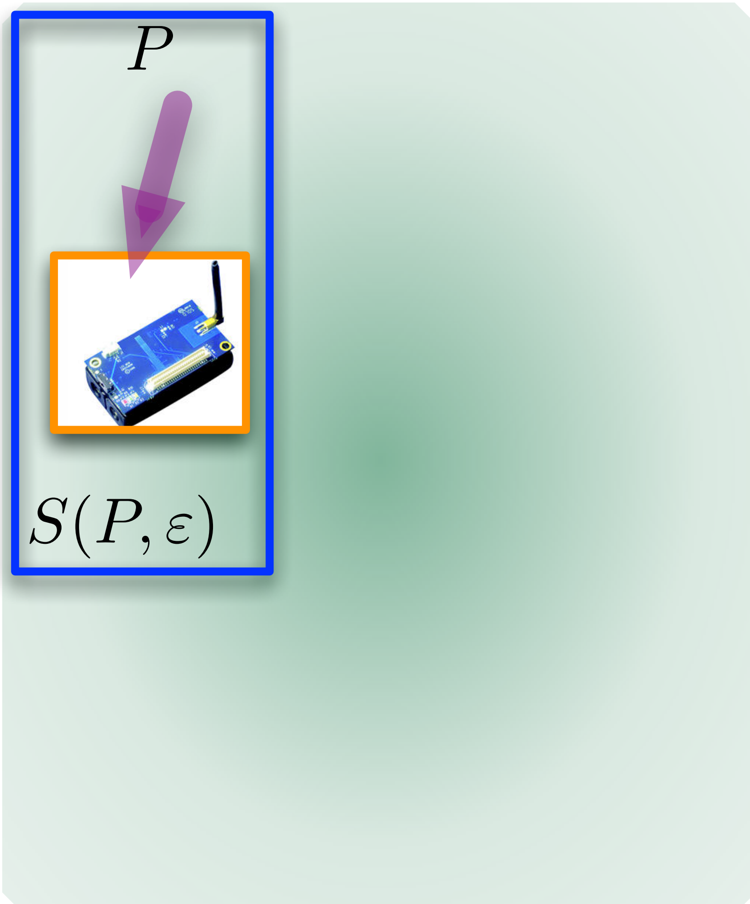
Heavy Hitters Summaries

Misra-Gries (MG) sketch of $P[1...U]$:

- Keep k (index,count) pairs
- If existing index arrives, update count
- If new index arrives, make new pair, or decrement all counts



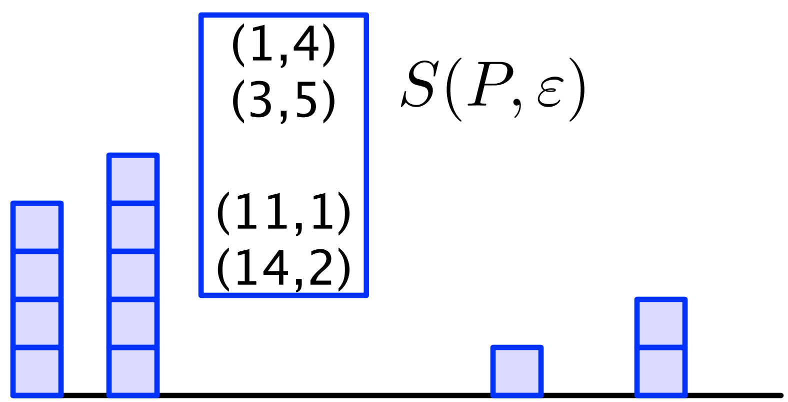
Heavy Hitters Summaries



Misra-Gries (MG) sketch of $P[1 \ldots U]$:

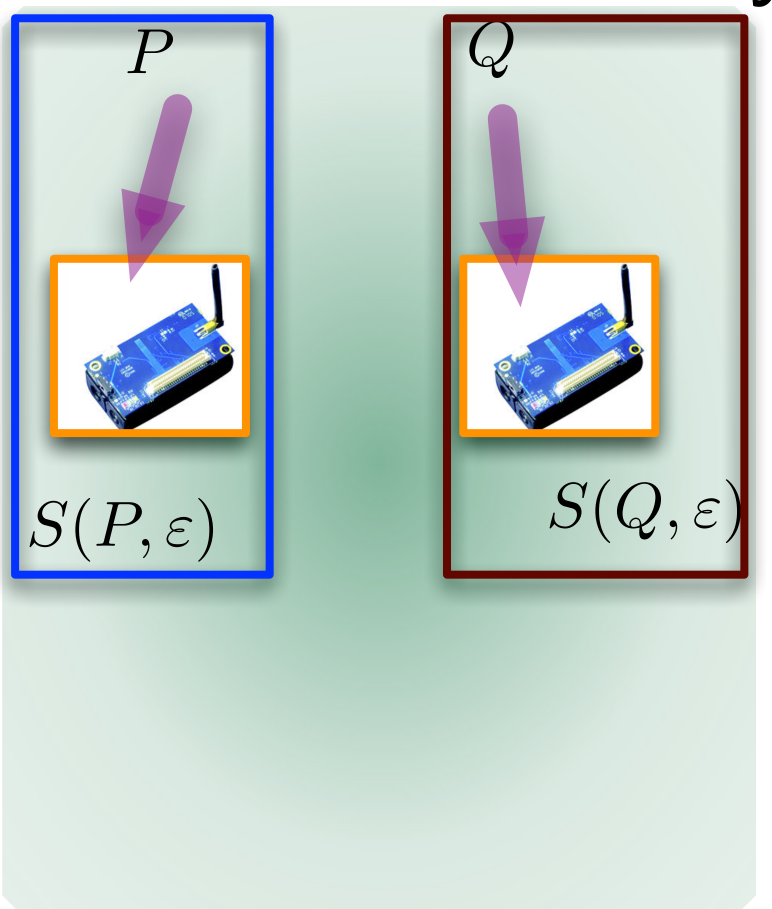
- Keep k (index,count) pairs
- If existing index arrives, update count
- If new index arrives, make new pair,
  or decrement all counts

$$|P[i] - \text{MG}[i]| \leq \varepsilon = \hat{m}/(k+1)$$



$S(P, \varepsilon)$

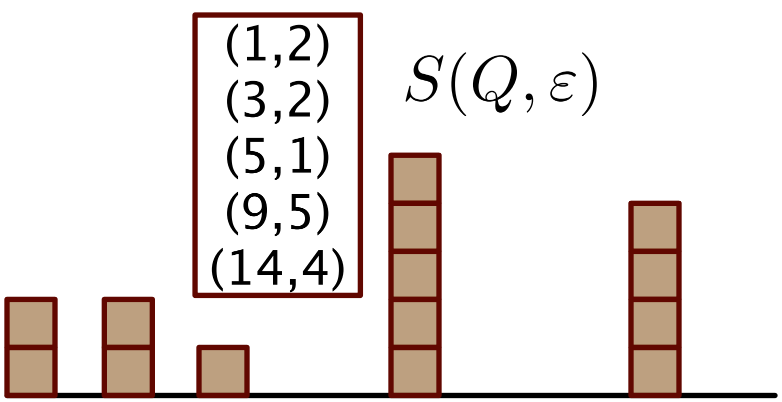
Heavy Hitters Summaries

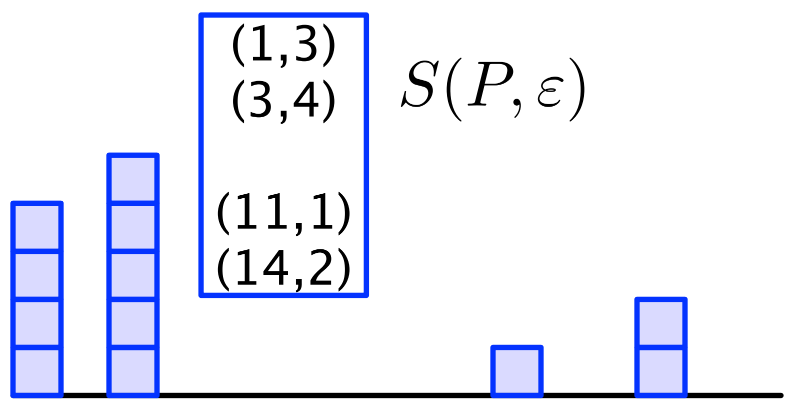


Misra-Gries (MG) sketch of $P[1...U]$:

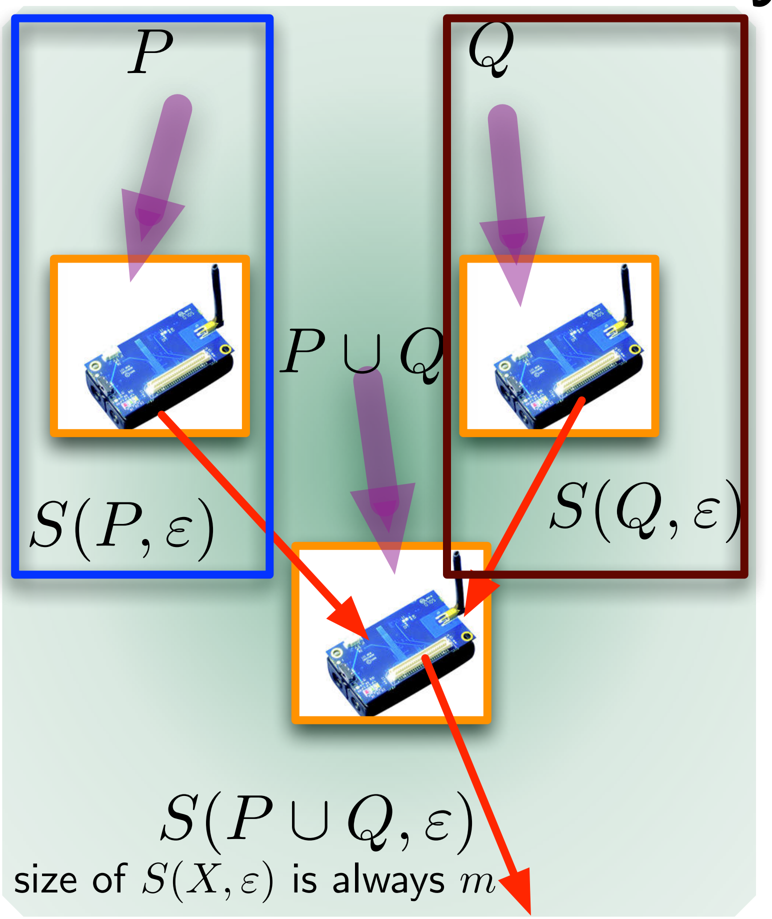
- Keep k (index,count) pairs
- If existing index arrives, update count
- If new index arrives, make new pair,
  or decrement all counts

Mergeable: Stack $\text{MG}(P) + \text{MG}(Q)$,





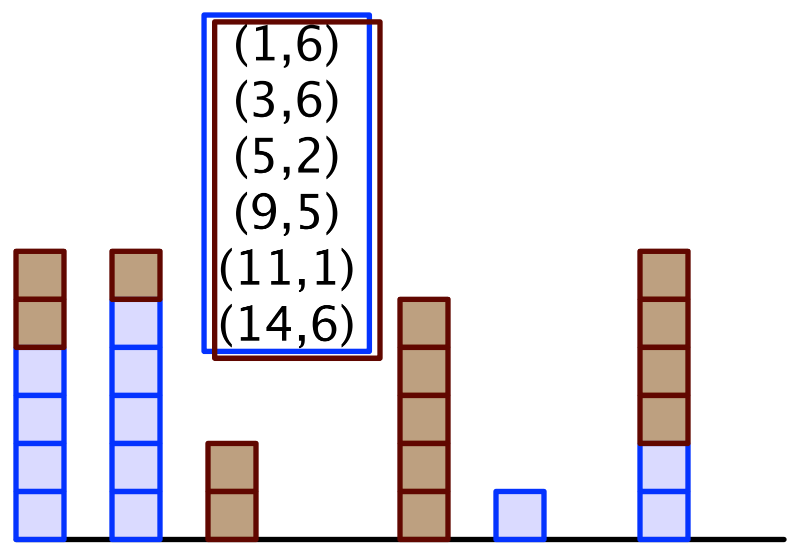
Heavy Hitters Summaries



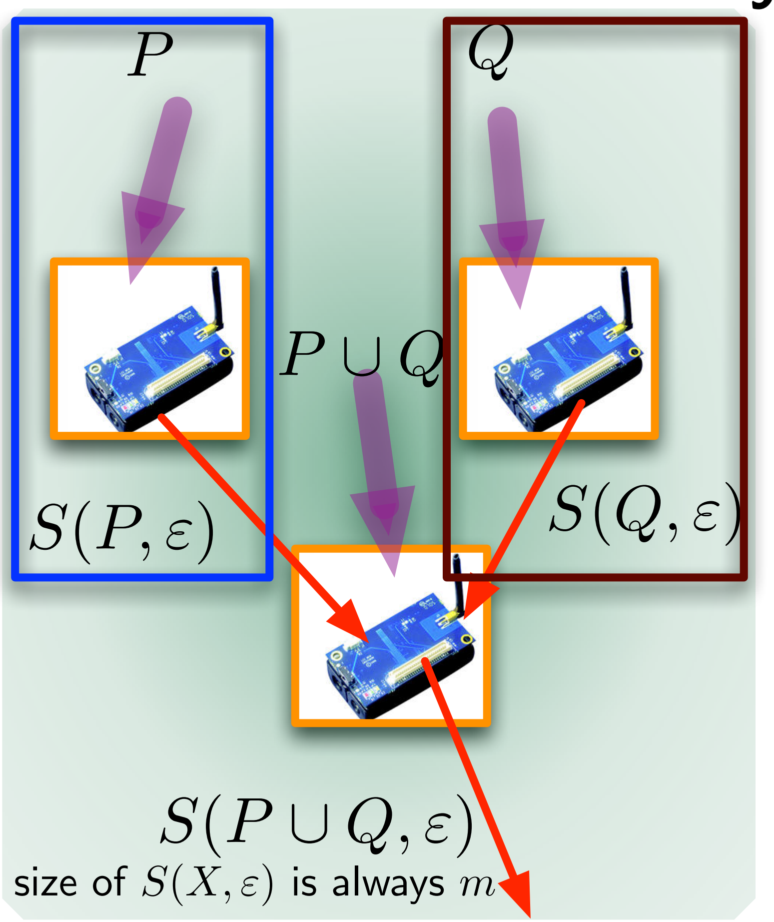
Misra-Gries (MG) sketch of $P[1...U]$:

- Keep k (index,count) pairs
- If existing index arrives, update count
- If new index arrives, make new pair, or decrement all counts

Mergeable: Stack MG(P) + MG(Q),



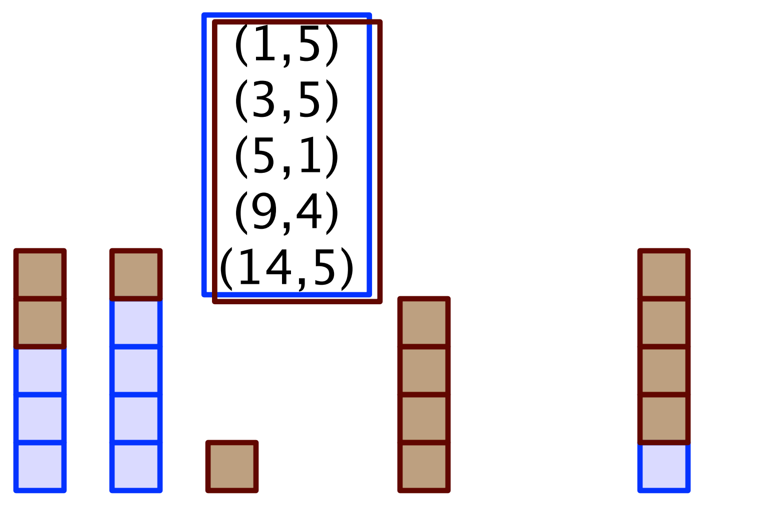
Heavy Hitters Summaries



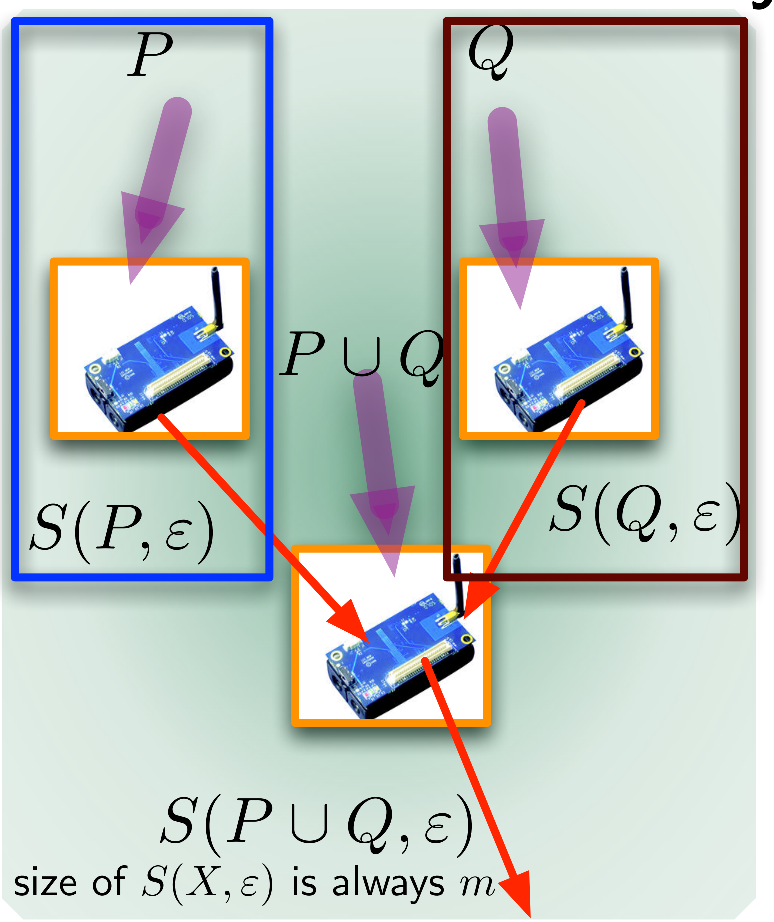
Misra-Gries (MG) sketch of $P[1...U]$:

- Keep k (index,count) pairs
- If existing index arrives, update count
- If new index arrives, make new pair, or decrement all counts

Mergeable: Stack MG(P) + MG(Q), decrement all counts C_{k+1}



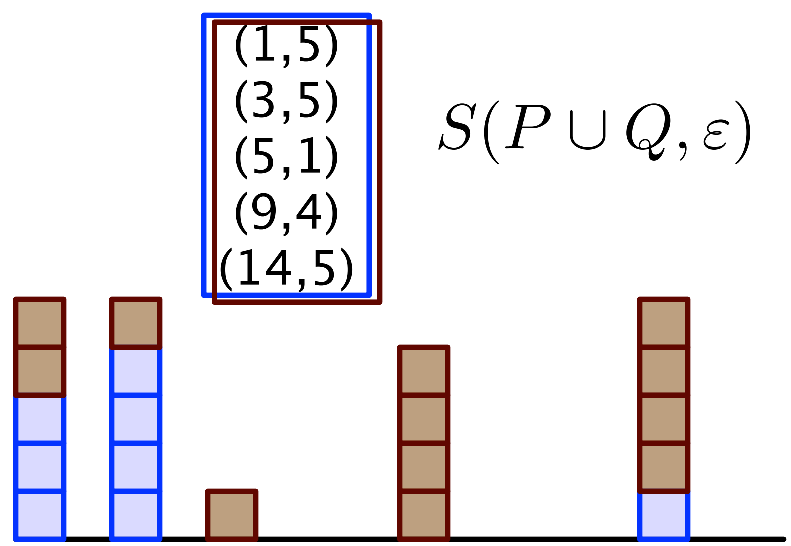
Heavy Hitters Summaries



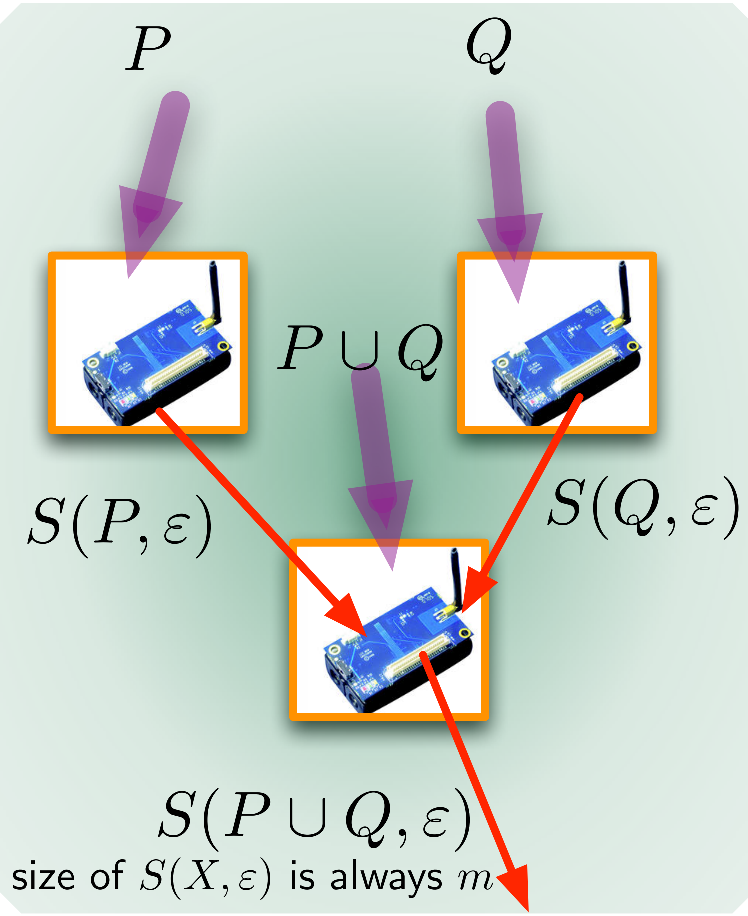
Misra-Gries (MG) sketch of $P[1...U]$:

- Keep k (index,count) pairs
- If existing index arrives, update count
- If new index arrives, make new pair, or decrement all counts

Mergeable: Stack MG(P) + MG(Q), decrement all counts C_{k+1}

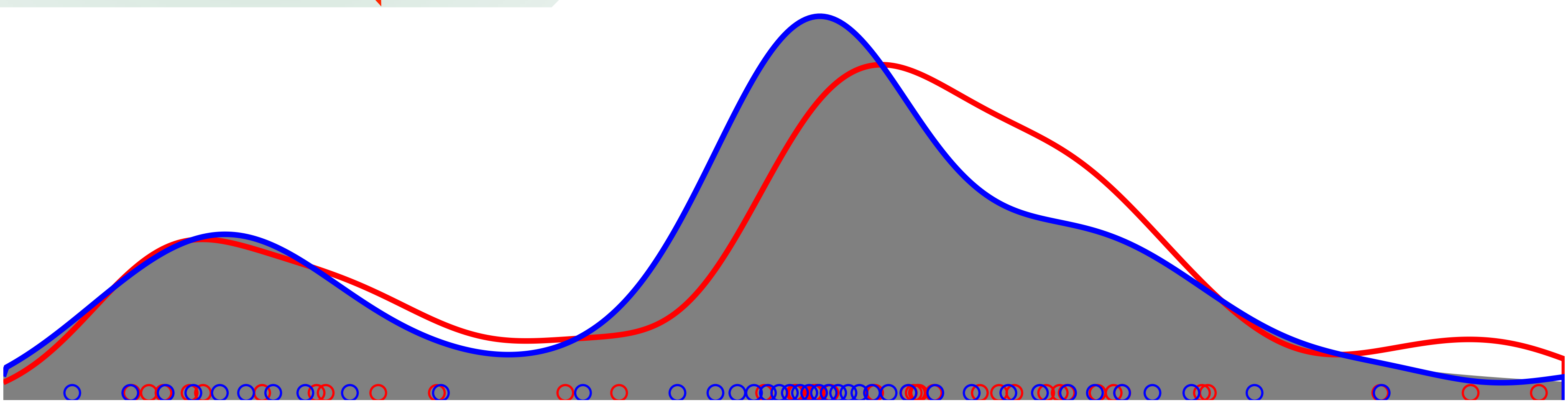


ε-Samples (Intervals)

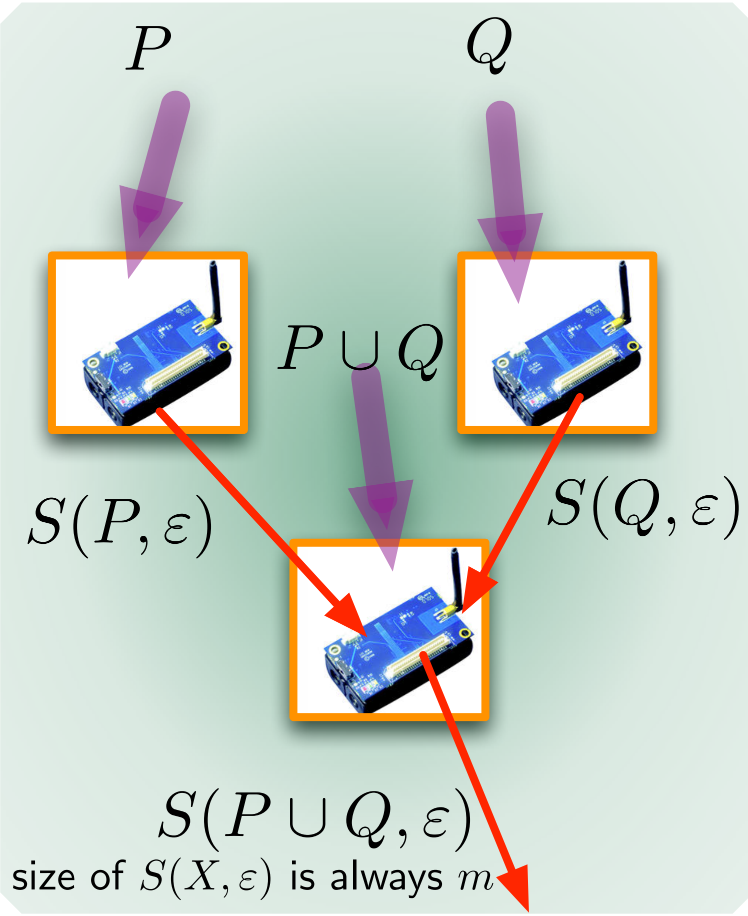


P

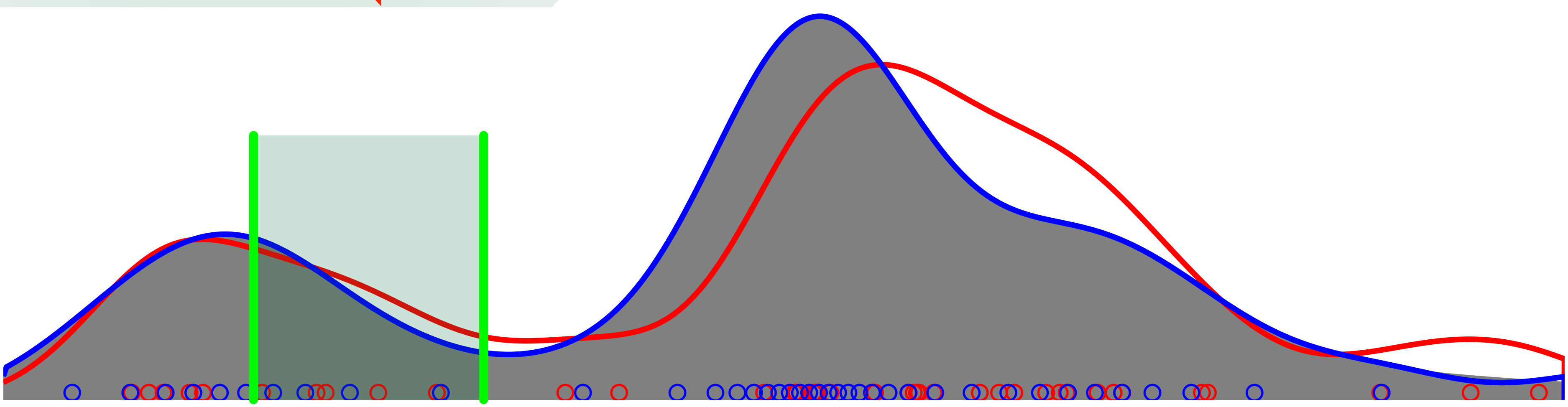
val	15	17	20	1	8	42	7	10	14	3



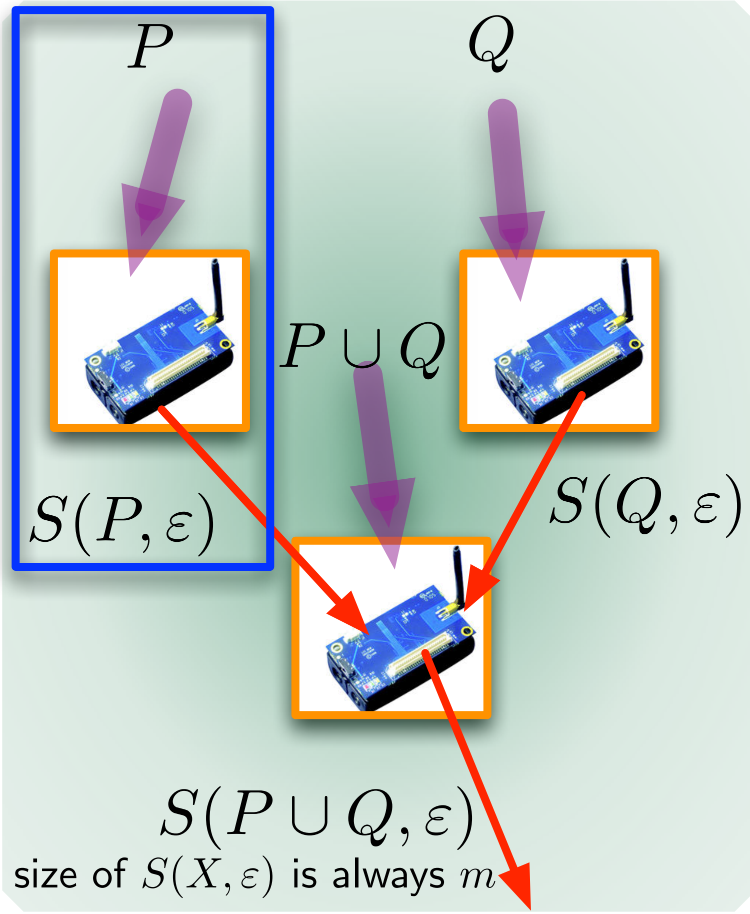
ε-Samples (Intervals)

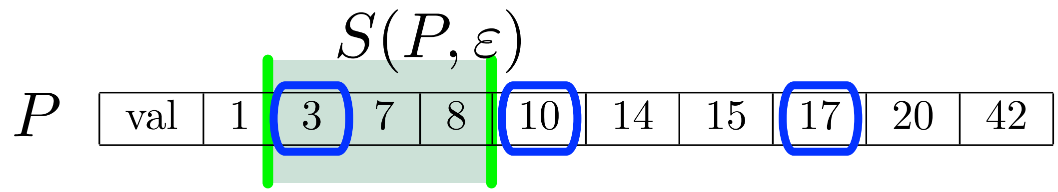


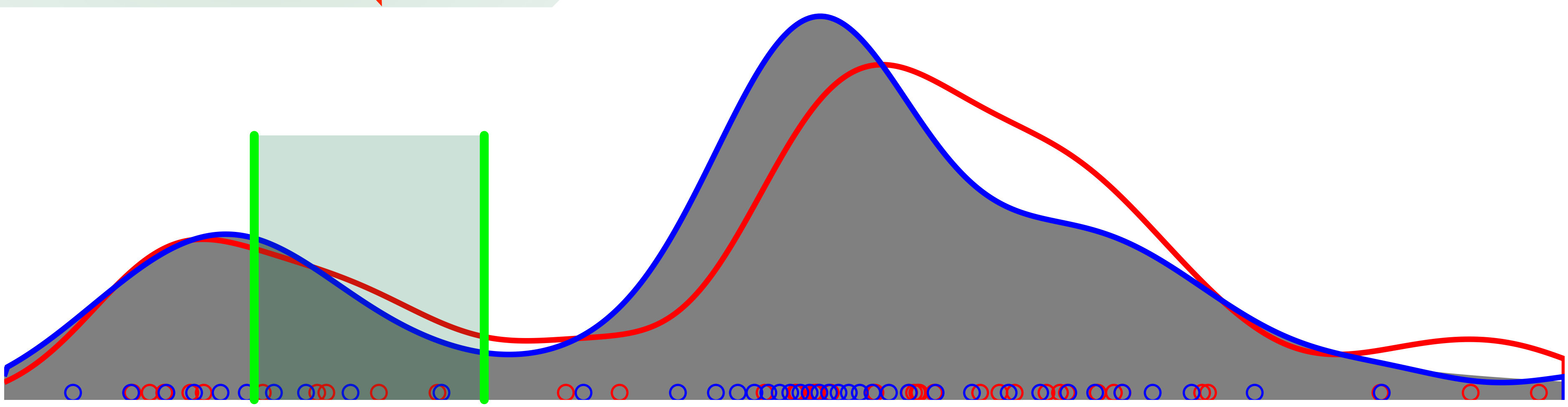
P	val	15	17	20	1	8	42	7	10	14	3



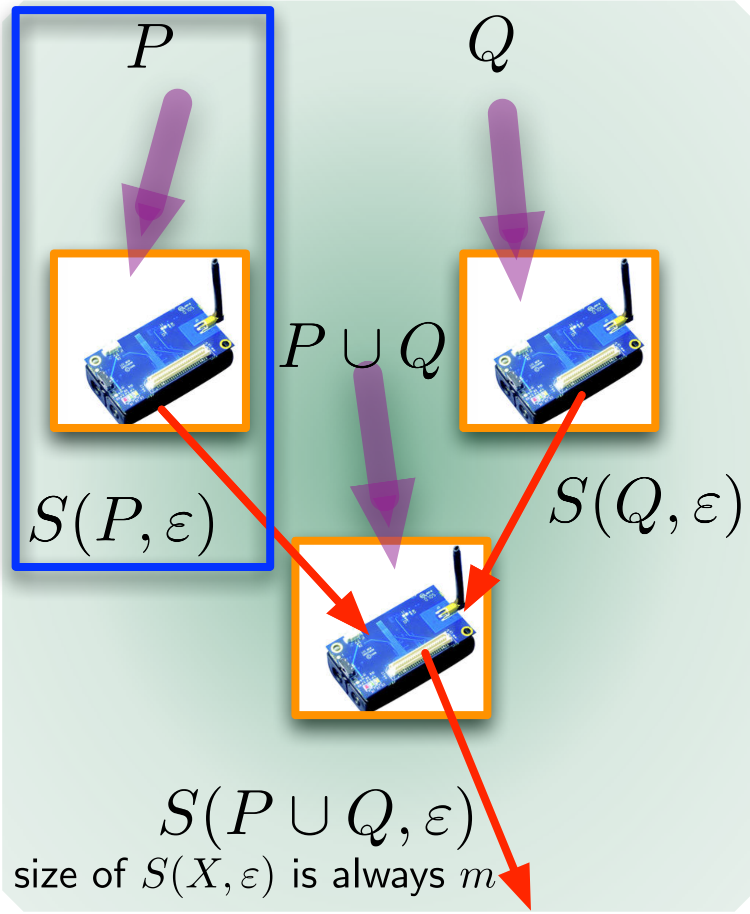
ε-Samples (Intervals)

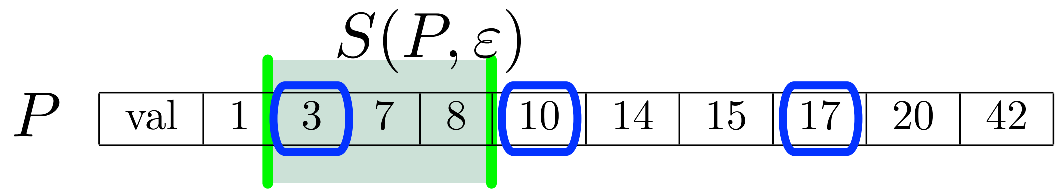




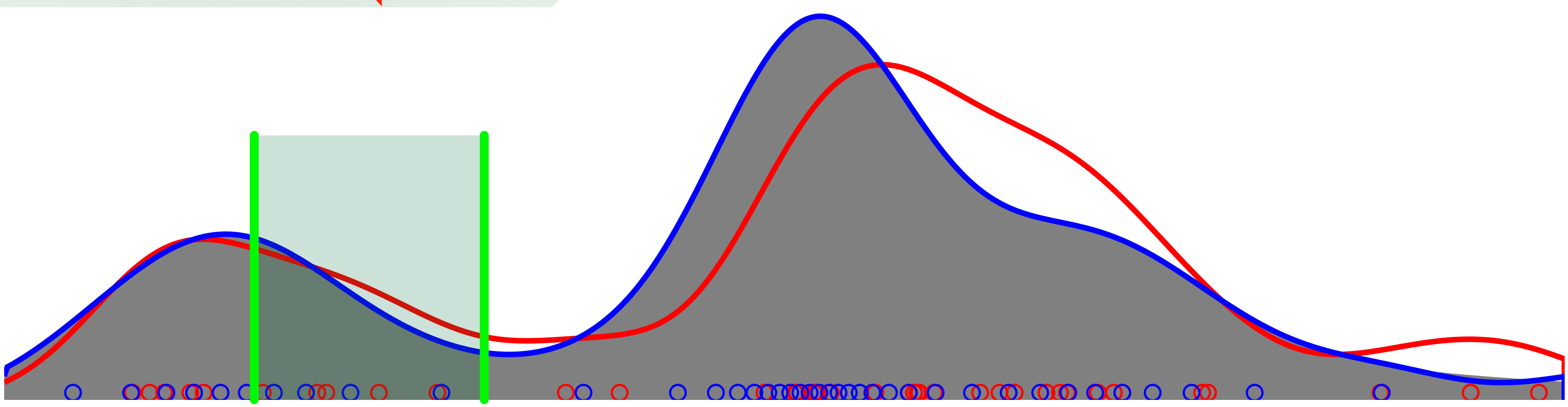


ε-Samples (Intervals)

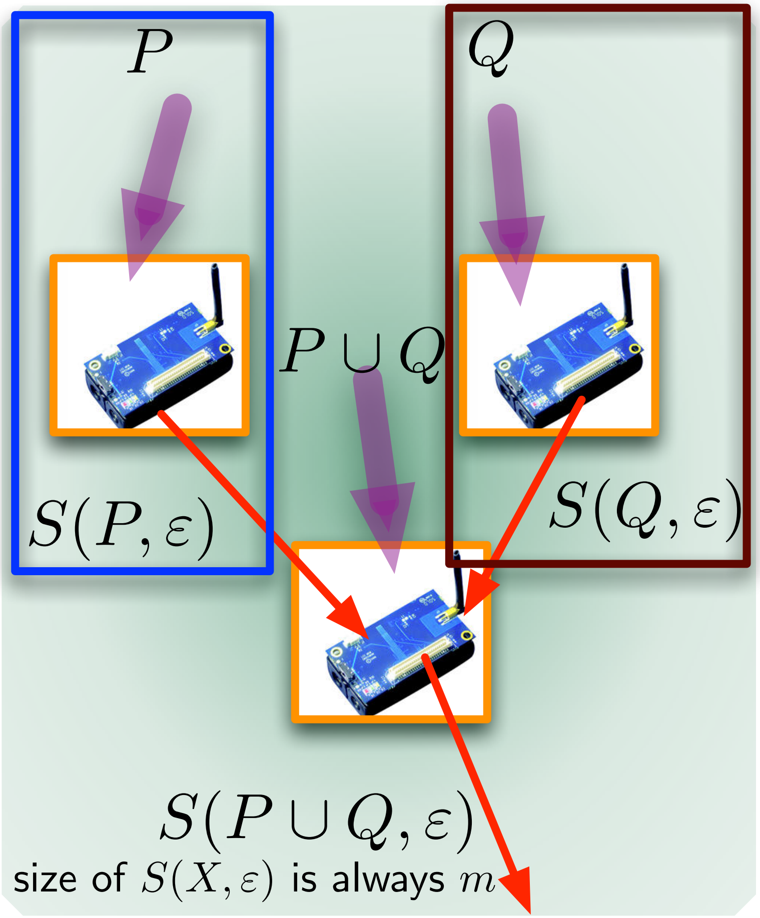




An ε-sample of ε-sample is a 2ε-sample



ε-Samples (Intervals)



$S(P, \varepsilon)$

P

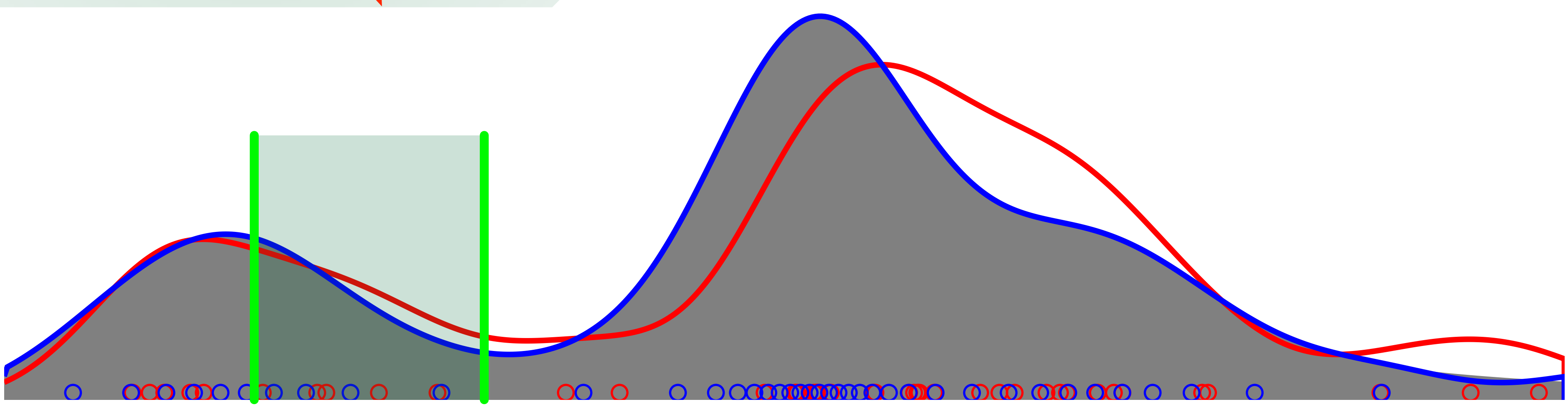
val	1	3	7	8	10	14	15	17	20	42

$S(Q, \varepsilon)$

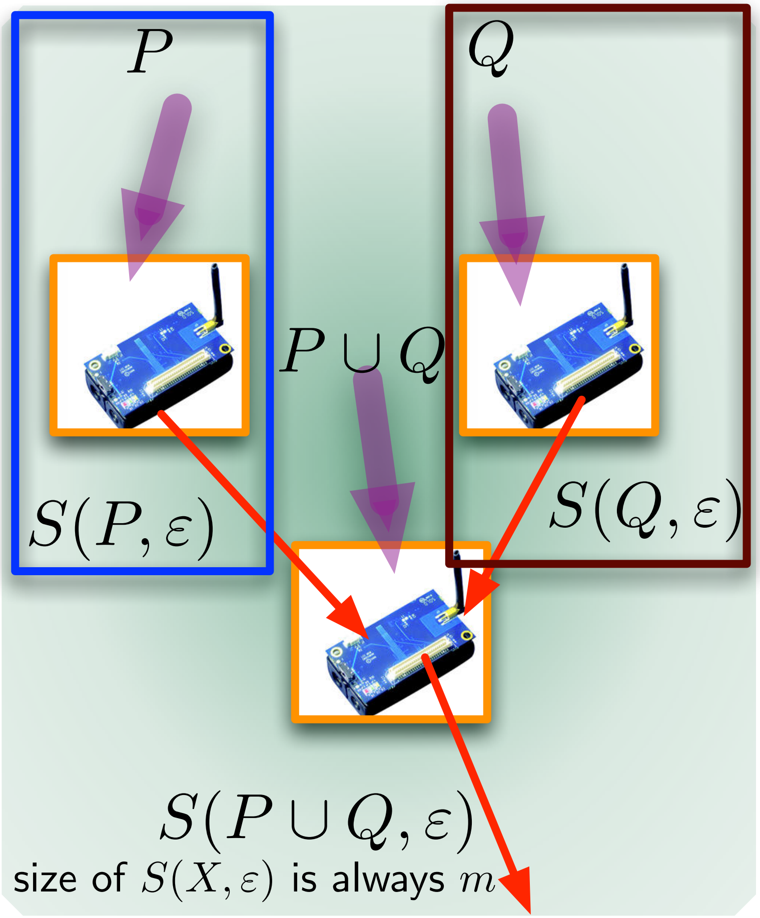
Q

val	2	4	7	9	11	13	14	16	21	31

val	3	4	10	11	16	17



ε-Samples (Intervals)



$S(P, \varepsilon)$

P

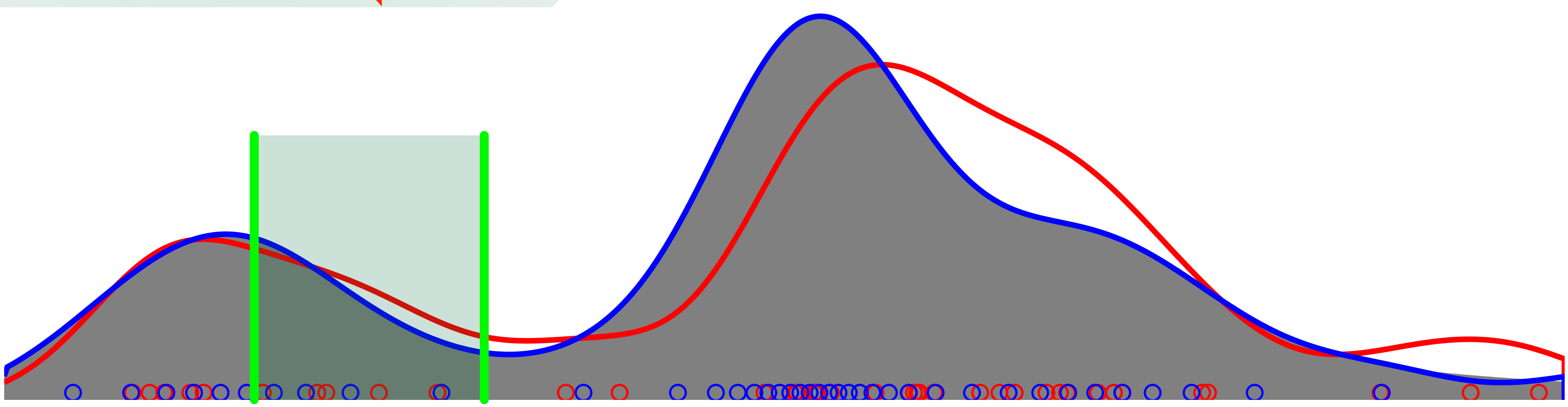
val	1	3	7	8	10	14	15	17	20	42

$S(Q, \varepsilon)$

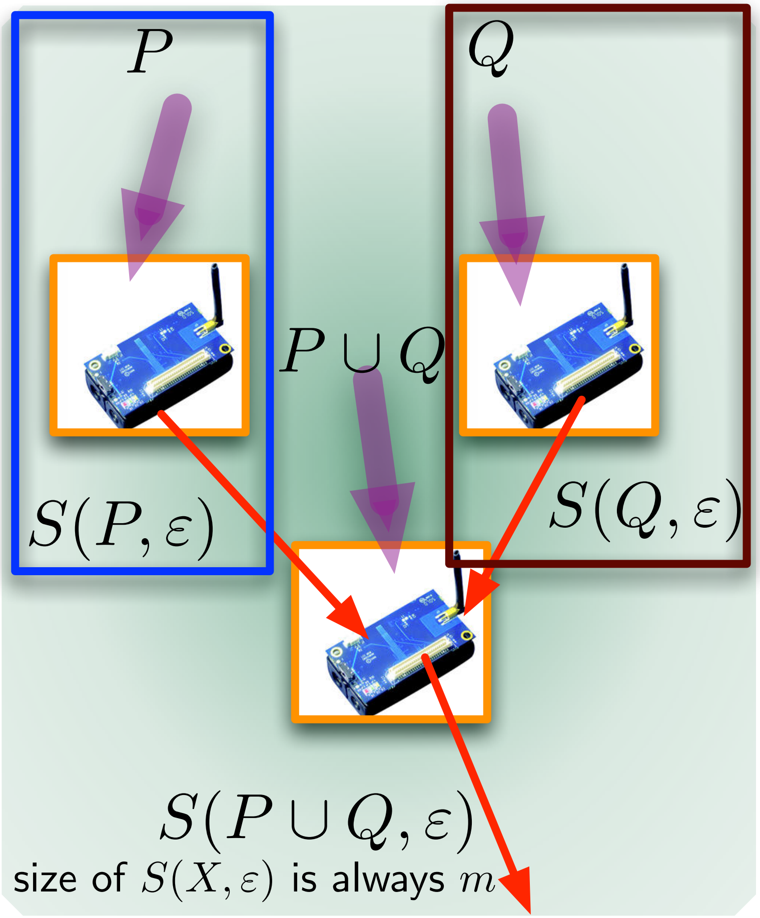
Q

val	2	4	7	9	11	13	14	16	21	31

val	3	4	10	11	16	17



ε-Samples (Intervals)



$S(P,\varepsilon)$

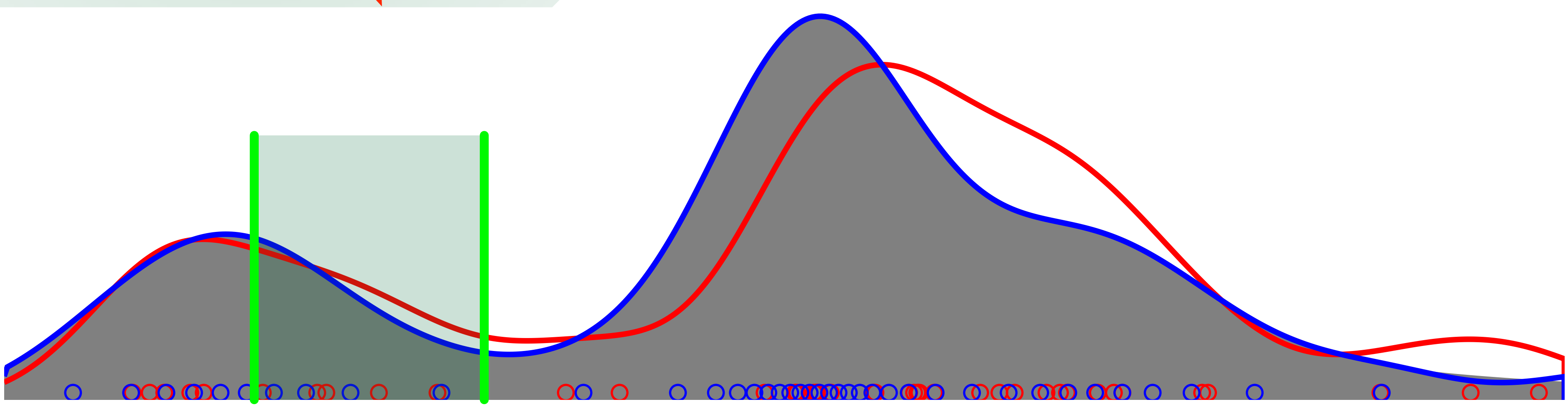
P	val	1	3	7	8	10	14	15	17	20	42

$S(Q,\varepsilon)$

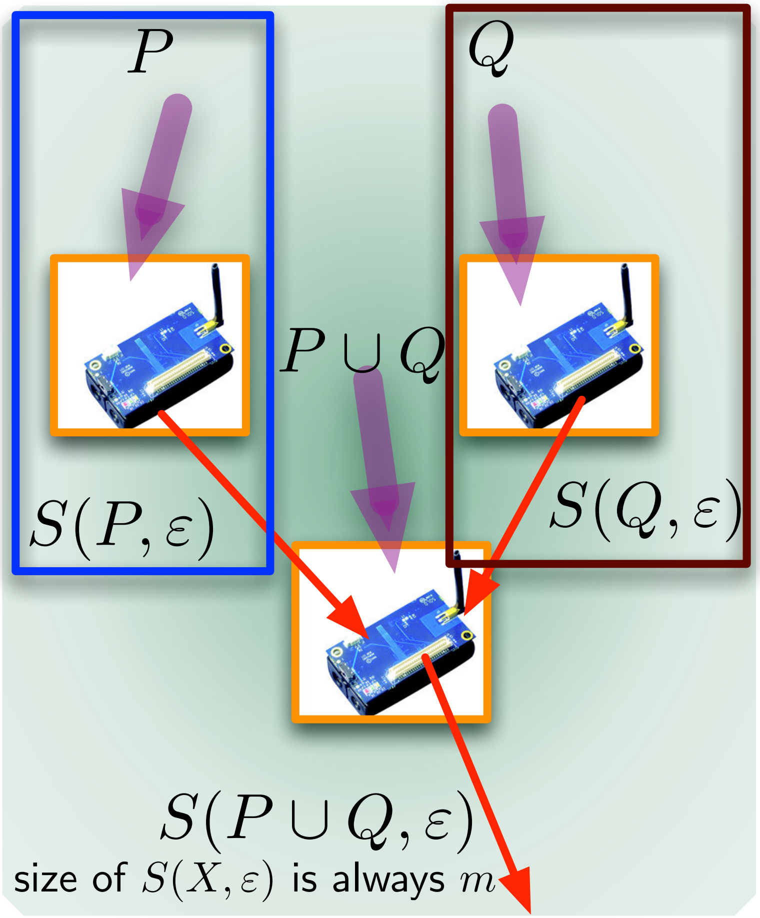
Q	val	2	4	7	9	11	13	14	16	21	31

$S(P\cup Q,\varepsilon)$

val	3	4	10	11	16	17



ε-Samples (Intervals)



$S(P,\varepsilon)$

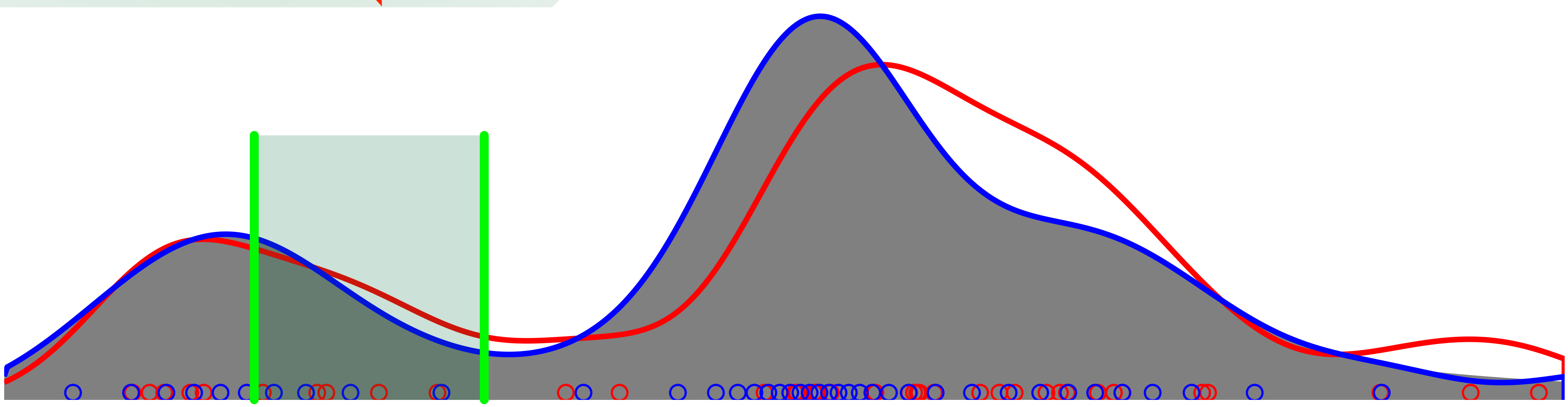
P	val	1	3	7	8	10	14	15	17	20	42

$S(Q,\varepsilon)$

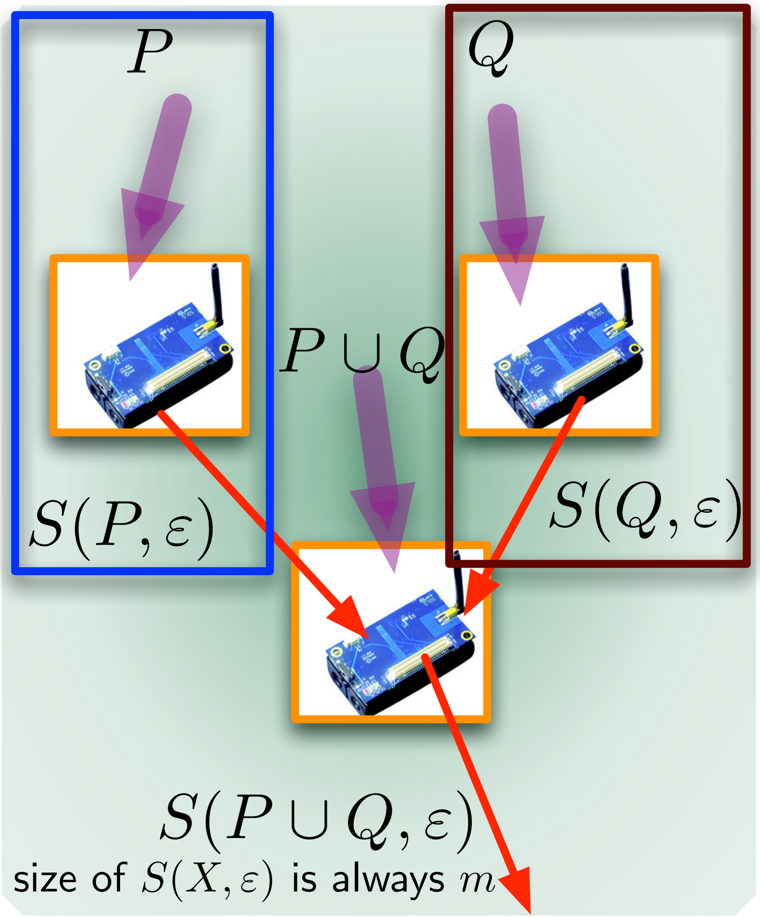
Q	val	2	4	7	9	11	13	14	16	21	31

$S(P \cup Q,\varepsilon)$

val	3	4	10	11	16	17



ε-Samples (Intervals)



$S(P, \varepsilon)$

P										
val	1	3	7	8	10	14	15	17	20	42

$S(Q, \varepsilon)$

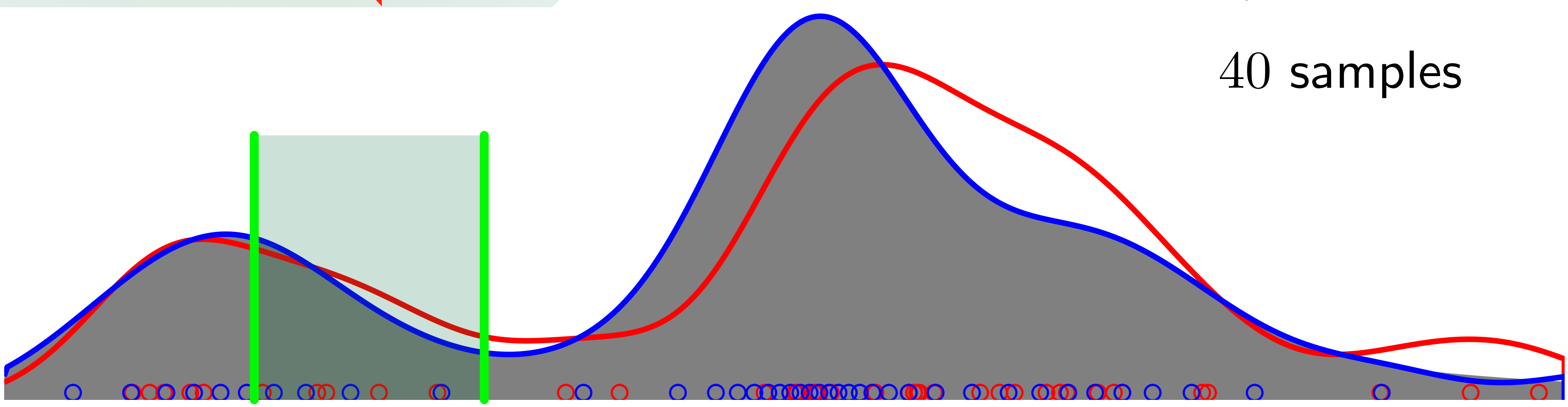
Q										
val	2	4	7	9	11	13	14	16	21	31

$S(P \cup Q, \varepsilon)$

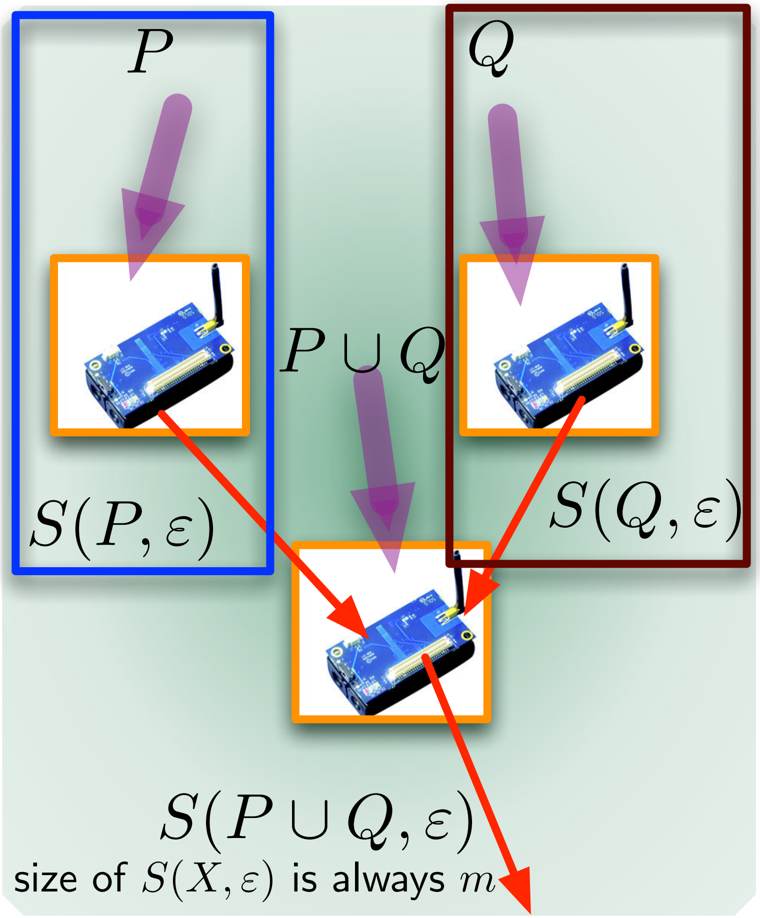
val	3	4	10	11	16	17

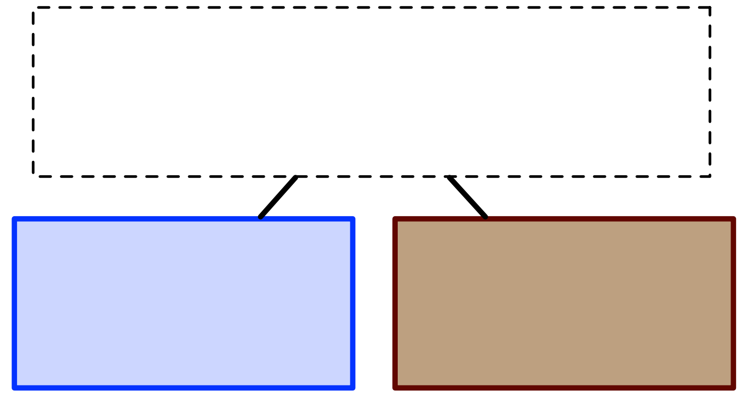
Random Sample: $(1/\varepsilon^2)\log(1/\delta)$.

Even-Weight Merge: $(1/\varepsilon)\sqrt{\log(1/\varepsilon\delta)}$.

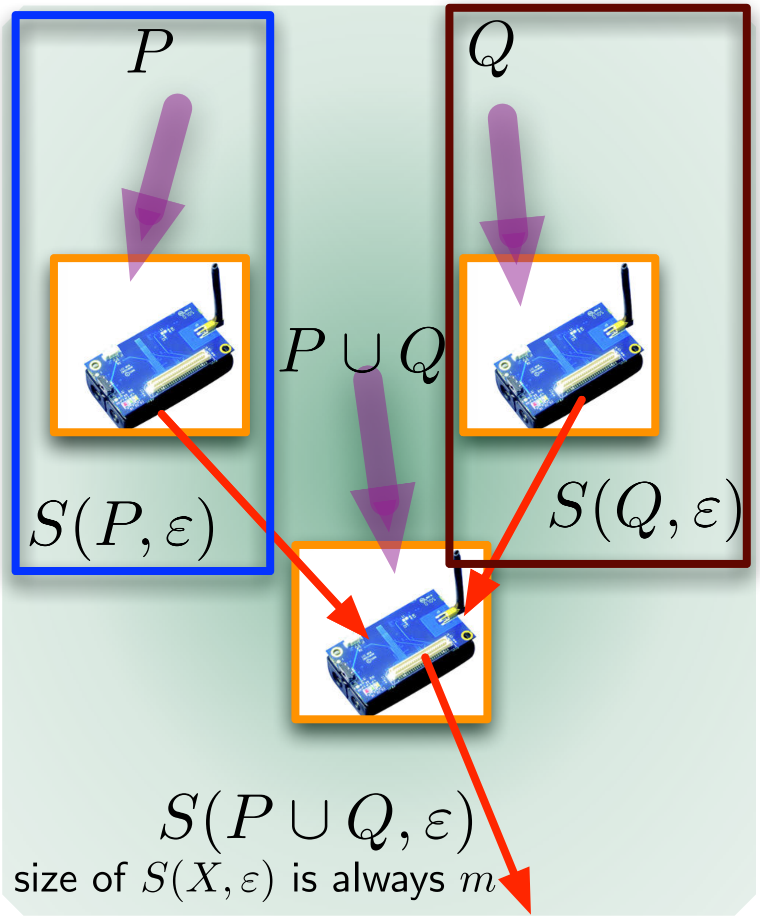


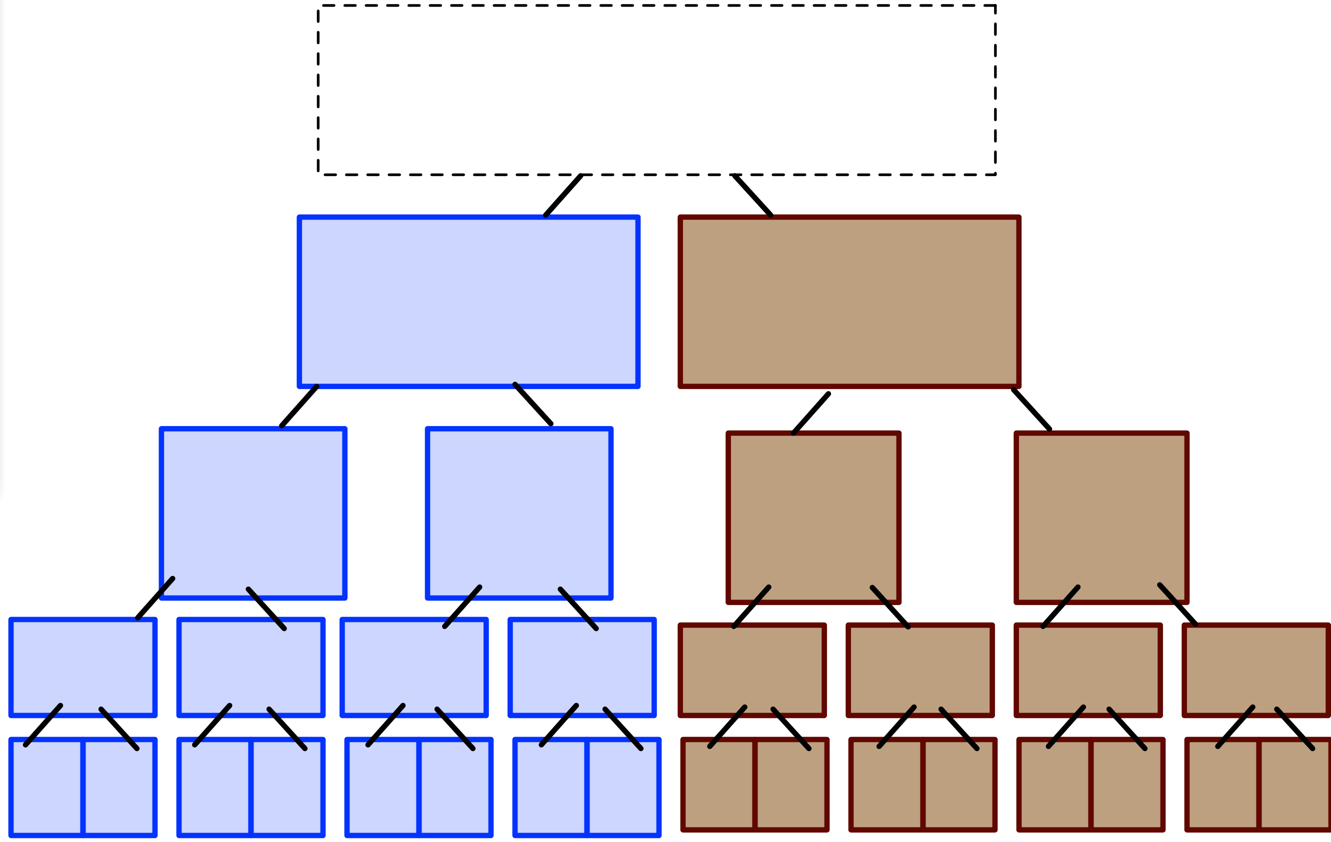
ε-Samples (Intervals)



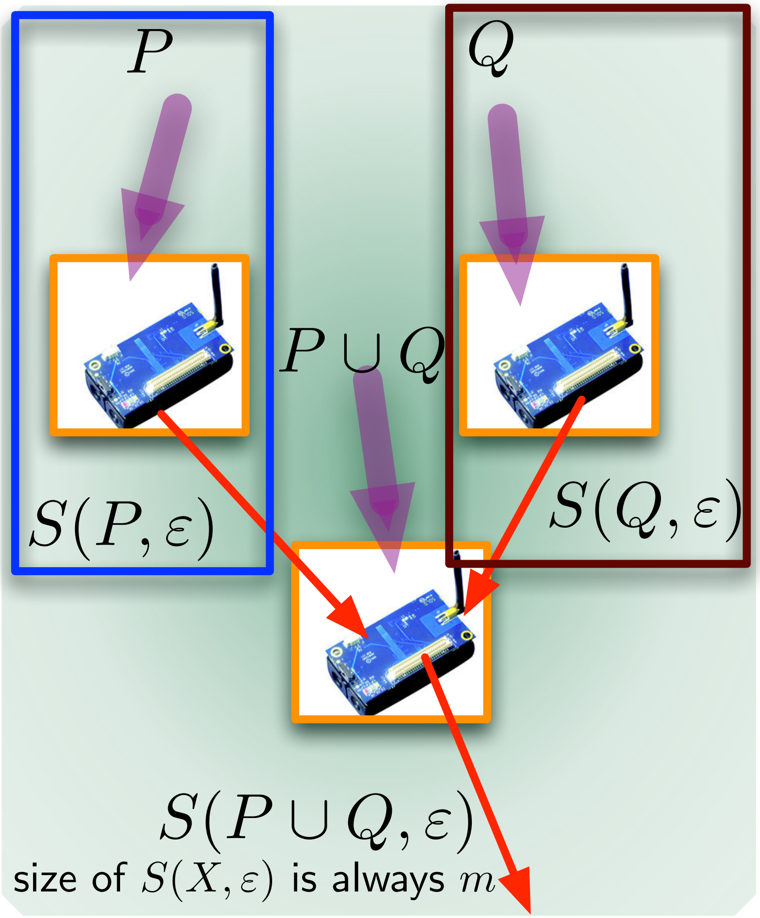


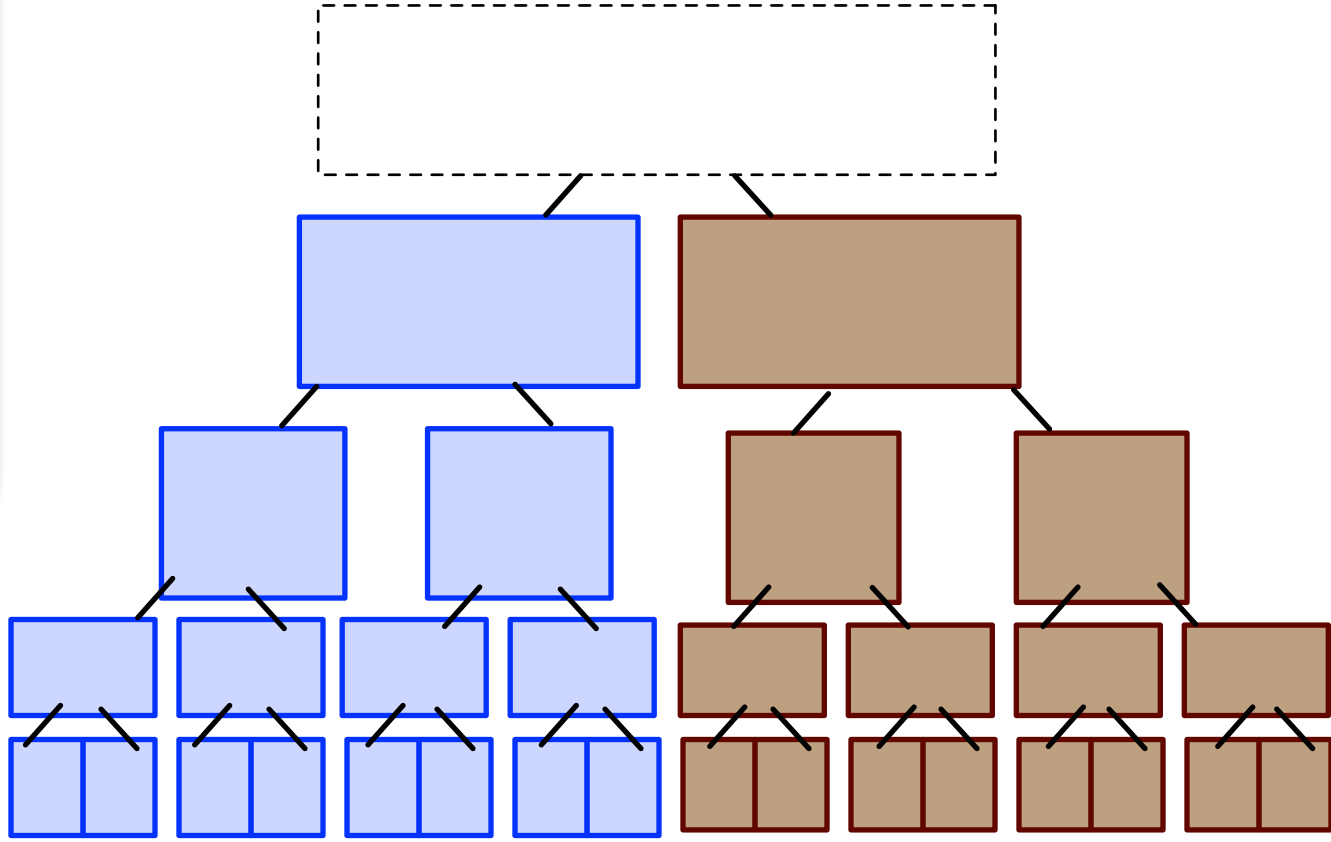
ε-Samples (Intervals)





ε-Samples (Intervals)

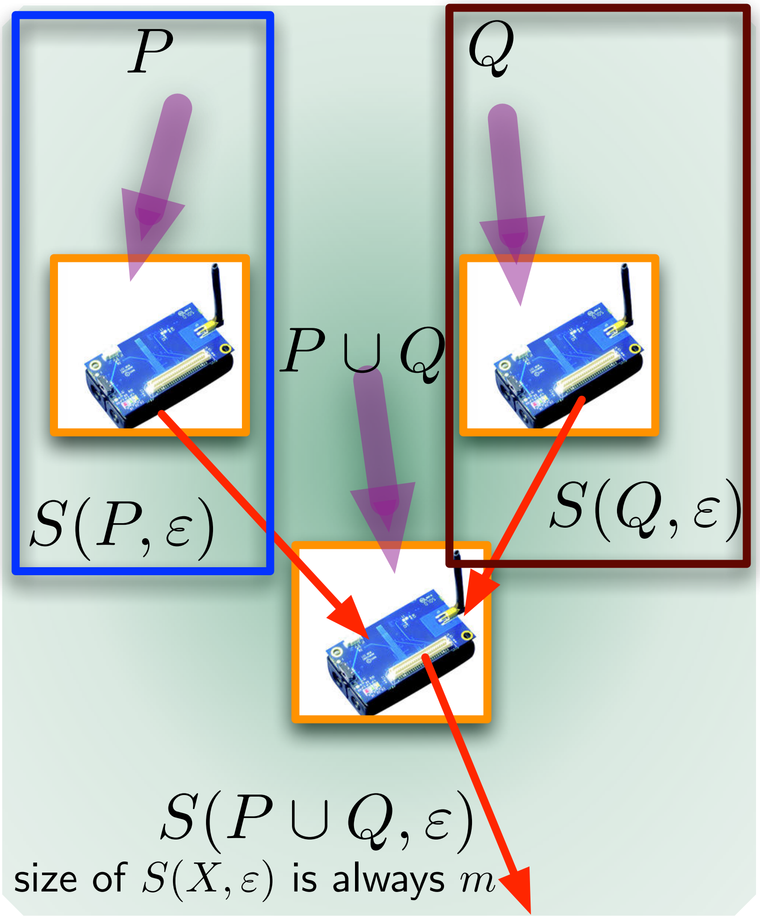


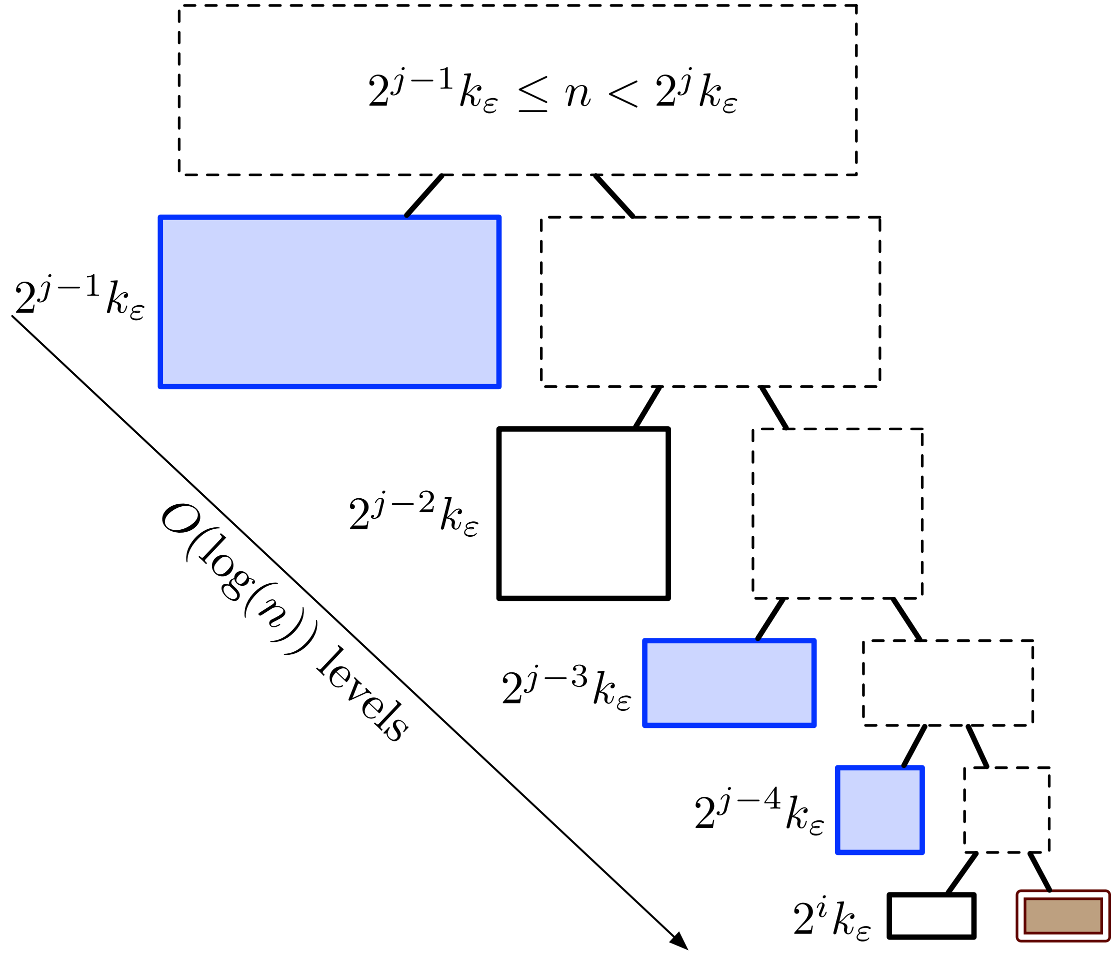


Let $E_{i,j}$ is jth merge error at level i.
$\mathbf{E}[E_{i,j}] = 0$ and $|E_{i,j}| \leq 2^i = \Delta_i$

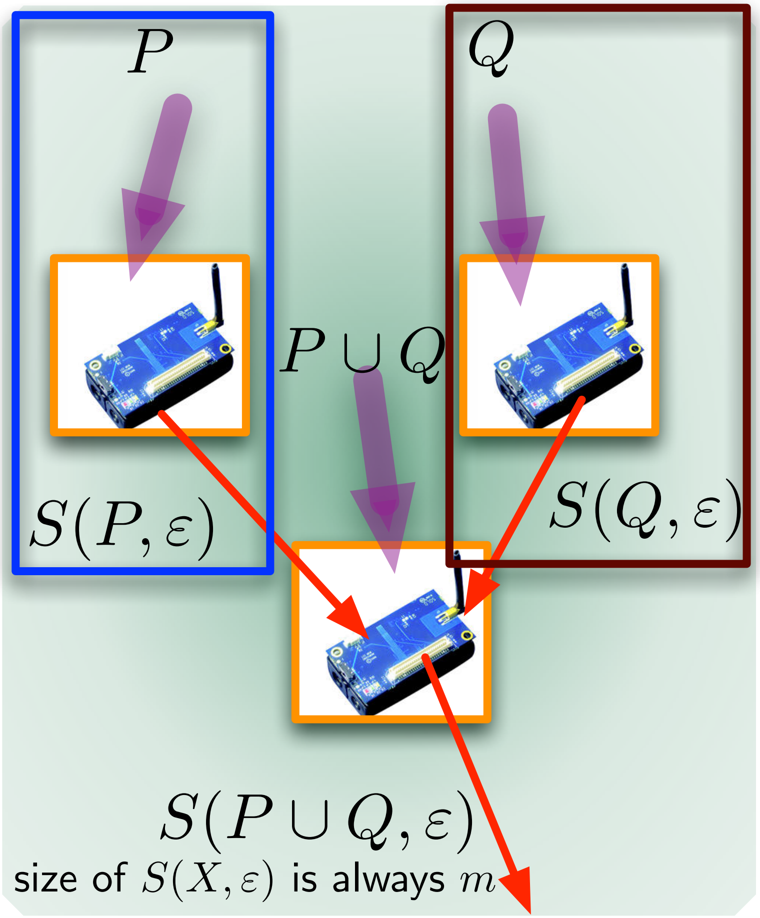
Chernoff-Hoeffding Bound:
$\mathbf{Pr}[\text{Err} > \varepsilon] \leq 2 \exp\left(\frac{-2\varepsilon^2}{\sum_i \sum_j \Delta_j^2}\right) \leq \delta$

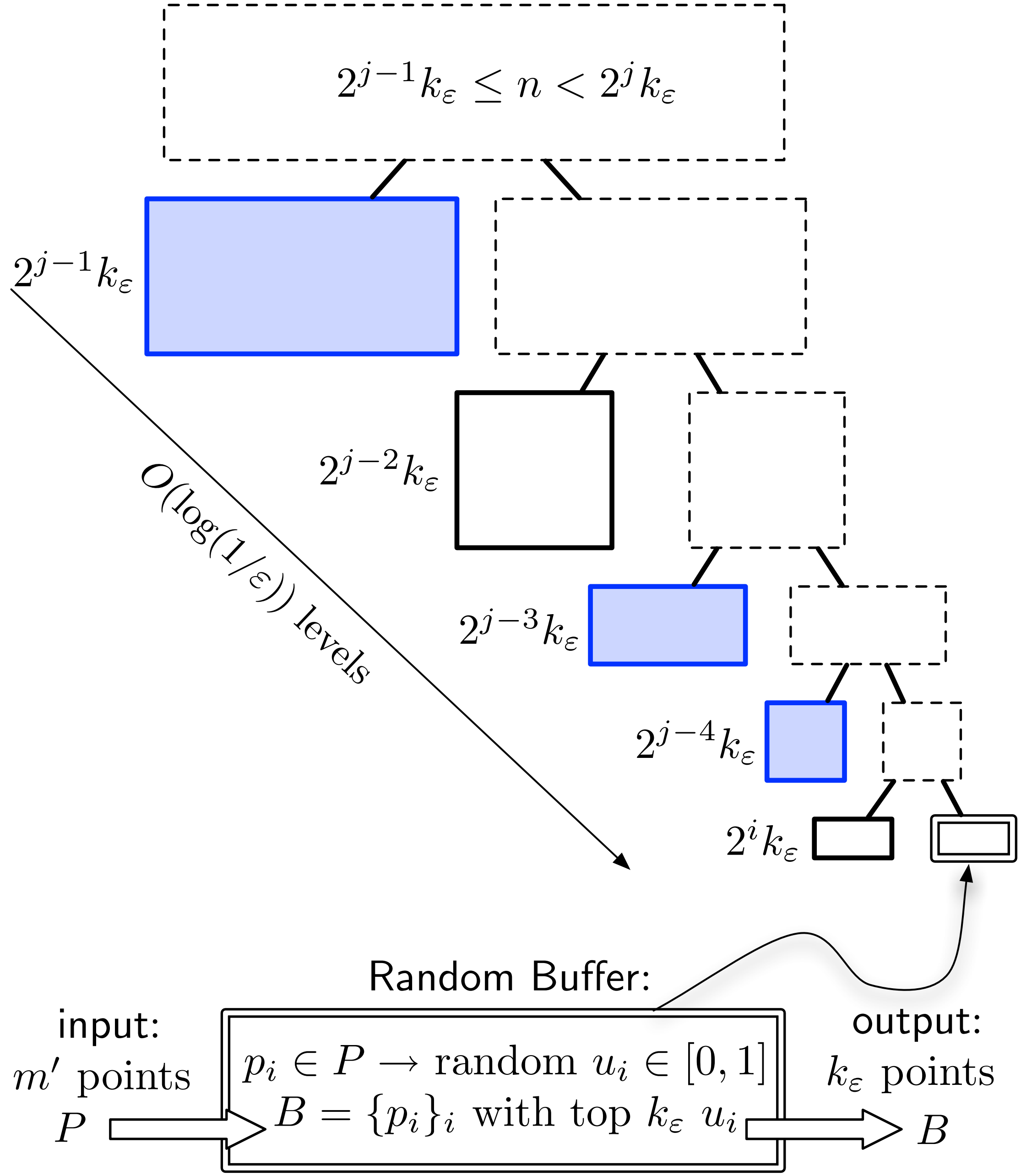
ε-Samples (Intervals)



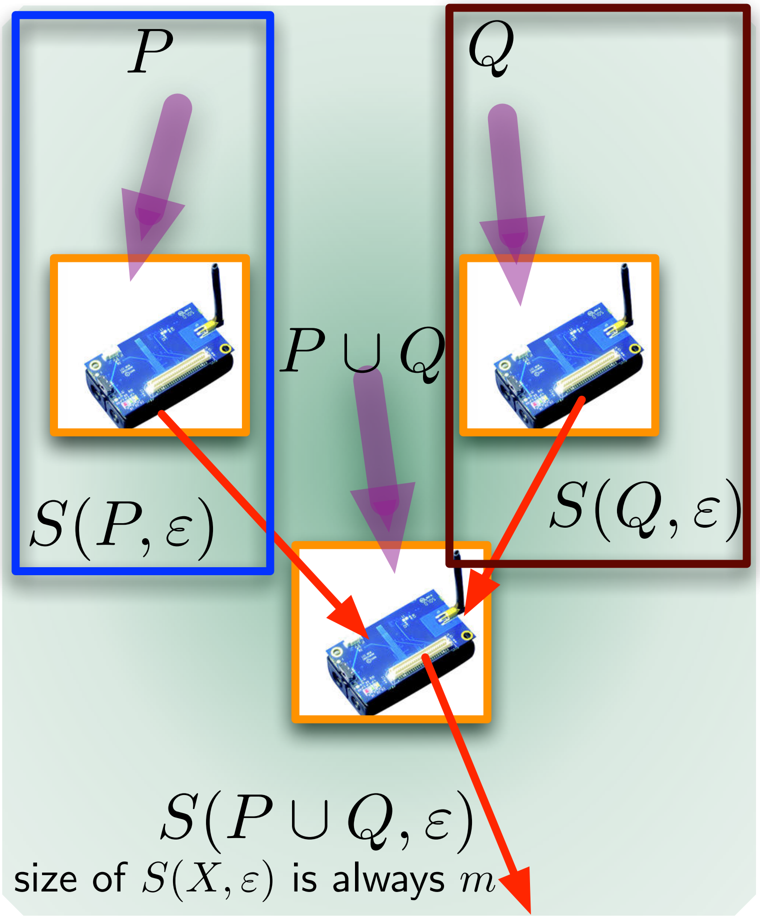


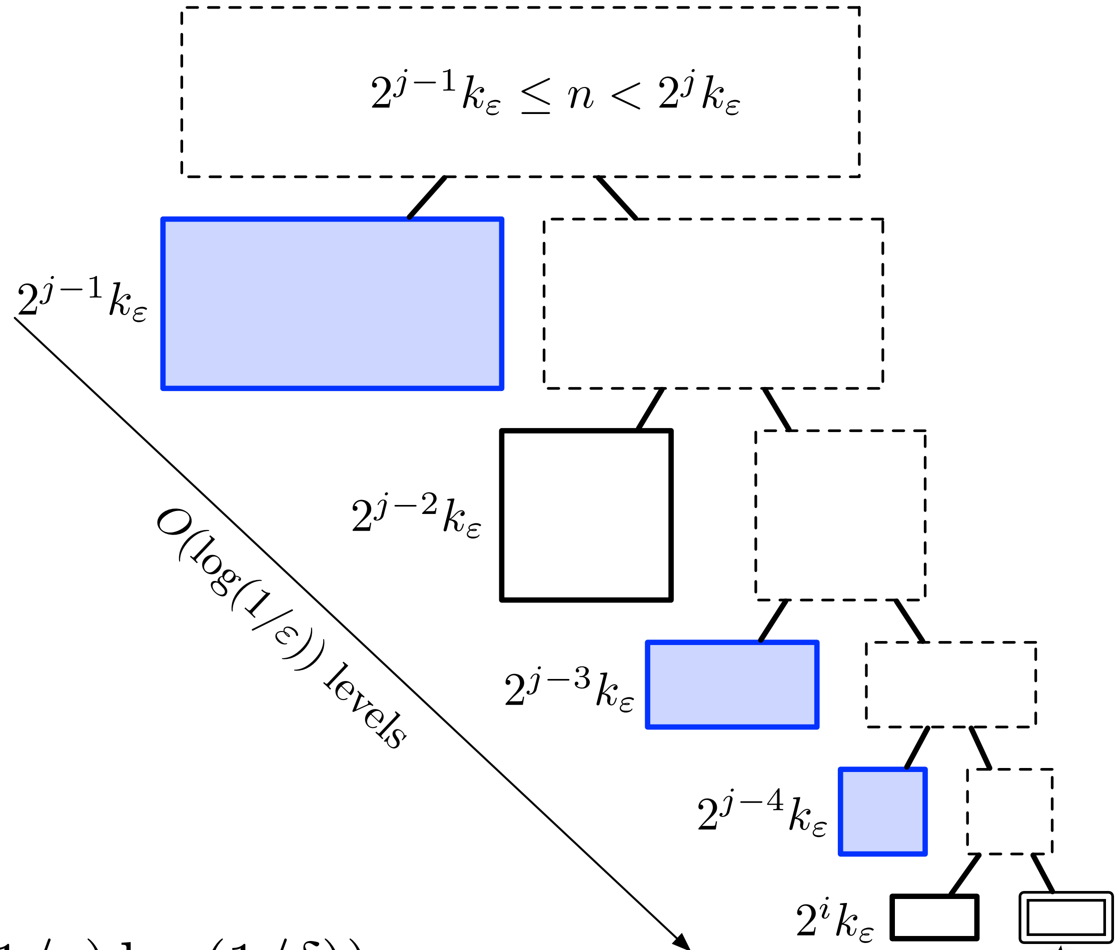
ε-Samples (Intervals)



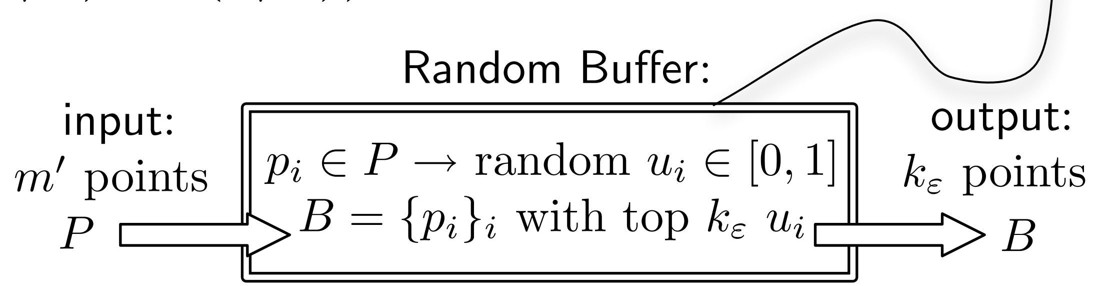


ε-Samples (Intervals)





$$m = O((1/\varepsilon) \log^{1.5}(1/\varepsilon) \log(1/\delta))$$

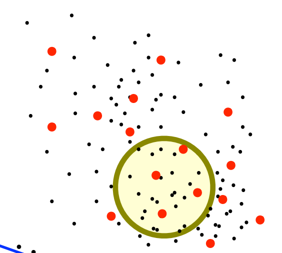


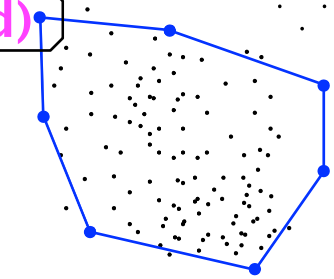
Mergeable Summaries for MASSIVE Data

Allows approximate computation with guarantees and small space

coreset: small summary, proxy for full data set
with approx guarantees:

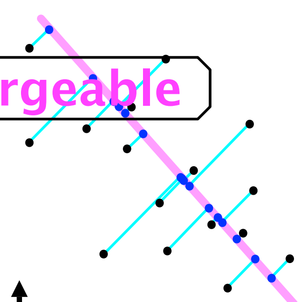
- ε-samples of $(P, \mathcal{R})$: approx density **Mergeable**
- ε-kernel: approx convex shape **Mergeable (restricted)**

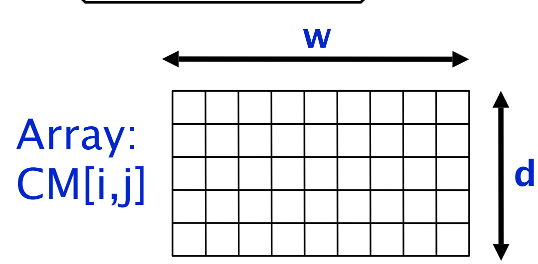




sketch: (random) (linear) combination of full data, recover functions
with approx guarantees:

- Euclidean distance: Johnson-Lindenstrauss random projection **Mergeable**
- min-count sketch: approx item counts **Mergeable**
- Greenwald-Khanna sketch: approx quantiles **One–way Mergeable**
- Misra-Gries sketch: approx frequent items **Mergeable**





Open Questions

- Mergeable ε-kernels without restrictions
- Mergeable summaries for clustering
- Mergeable summaries for PCA
- Mergeable summaries for graphs [next talk]
- Lower bounds for mergeable summaries (deterministic)
- Implementation Studies