5th meeting of the PRIN project “Multicriteria data structures”

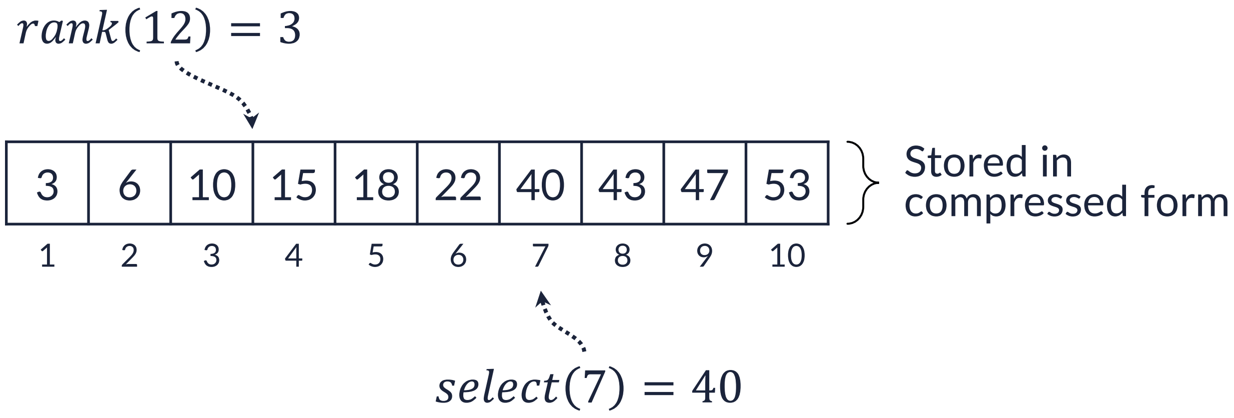
Advances on learned indexing and compression of integer data

Giorgio Vinciguerra

Università di Pisa

Compressed rank/select dictionaries

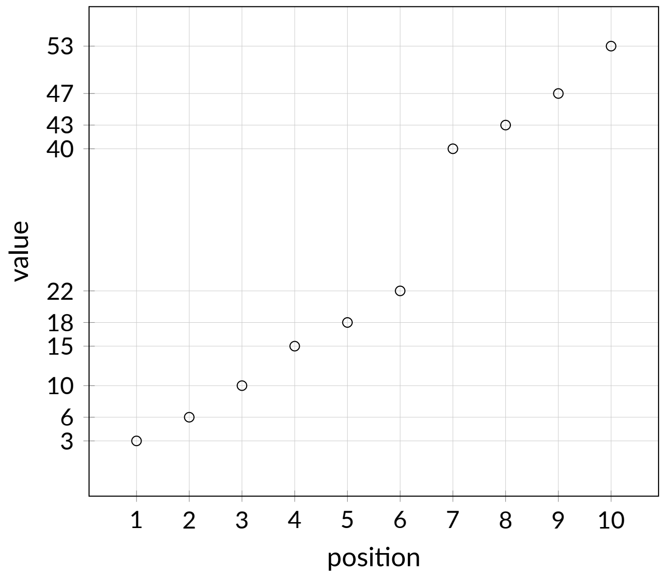
- Given a set A of n elements over an integer universe $0,1,\dots,u$
    1. Store them in compressed form
    2. Implement $rank(x)$: number of elements in A which are $\leq x$
    3. Implement $select(i)$: return the ith smallest element in A
- Well-studied building block of succinct data structures



Recap

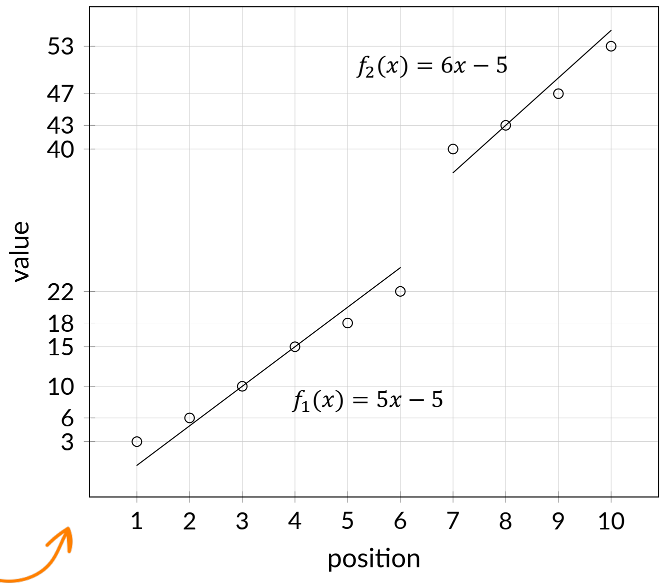
In ALENEX ’21 and TALG 2022, we showed how to use piecewise linear ε-approximations (**PLAs**) to build compressed R/S dictionaries

Step 1: map input to (pos, value)

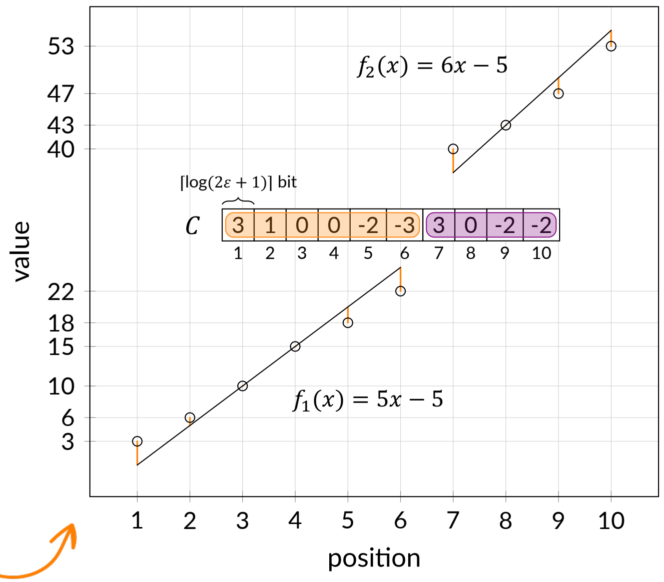


3	6	10	15	18	22	40	43	47	53
1	2	3	4	5	6	7	8	9	10

Step 2: build a PLA with error ε



Step 3: store segments + corrections

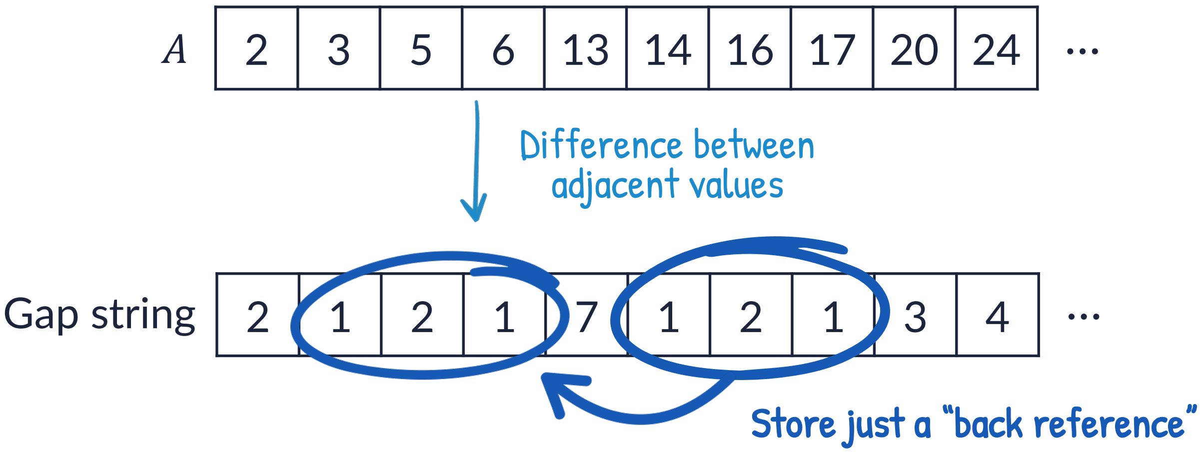


Crucial ingredient: minimising the space of the encoding via a PLA that uses a different ε for different segments

Joint work with Antonio and Paolo

Recap (cont.)

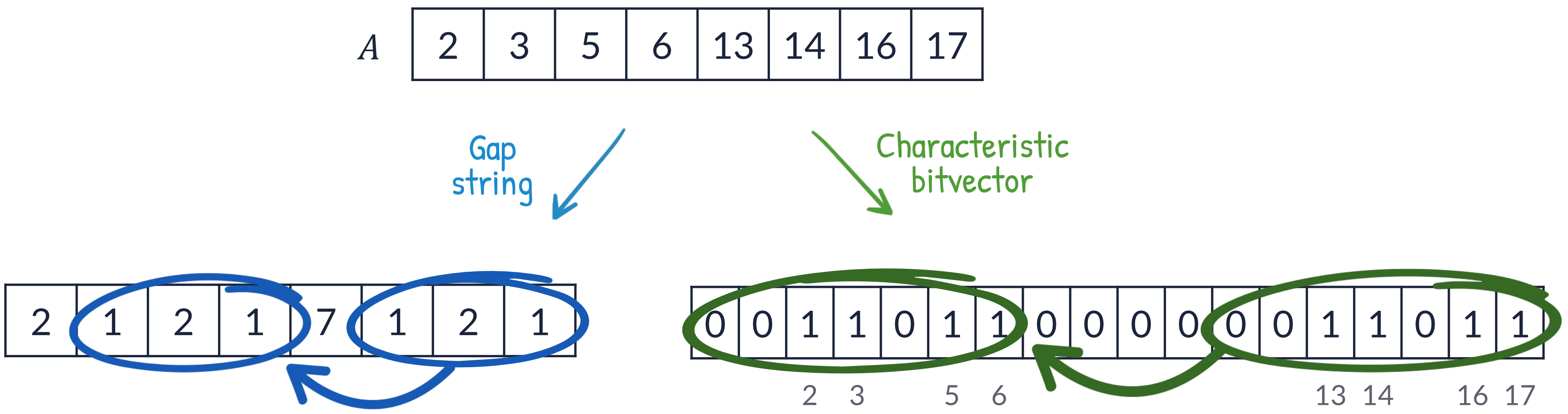
- In ISAAC ‘21 we showed how to combine PLAs with tools to capture repetitiveness



- Introduced and experimented the block-ε tree
- Introduced the LZ_ε parsing

New results (submitted to journal)

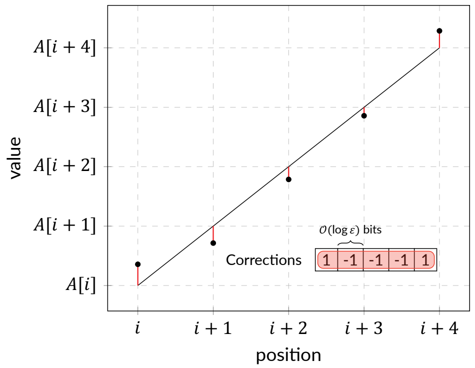
1. Gap vs binary entropy inequality

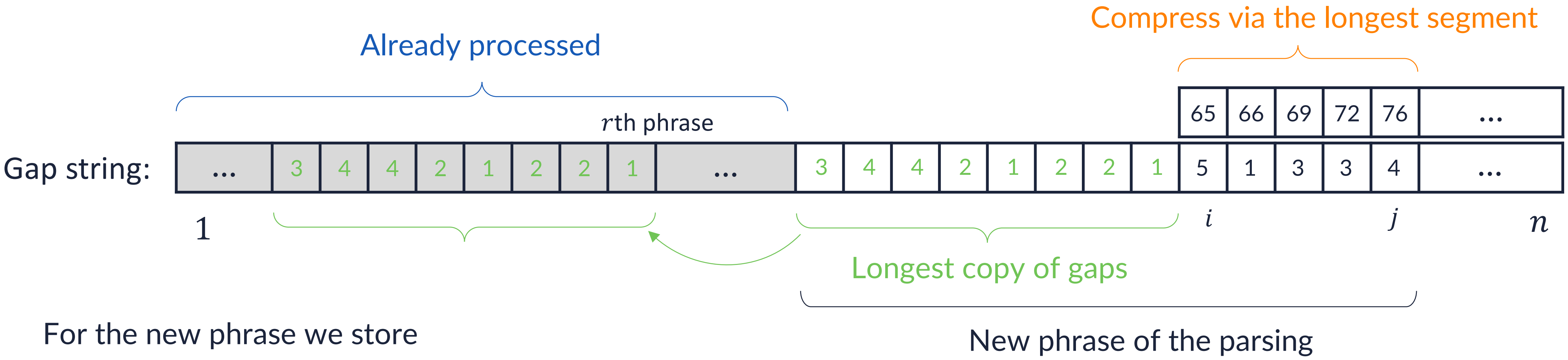


$$nH_k(\text{gap string}) \;\leq\; uH_k(\text{characteristic bitvector})$$

New results (submitted to journal)

2. Improved the LZ_ε parsing space bounds

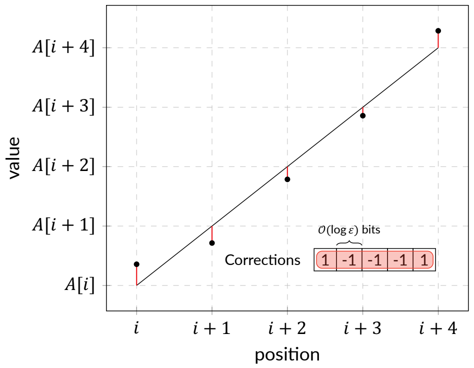




For the new phrase we store

- Indexes i, j, and r
- Slope and intercept of the segment
- Array of $j - i + 1$ corrections, $\lceil \log(2\varepsilon + 1) \rceil$ bits each

New results (submitted to journal)



2. Improved the LZ_ε parsing space bounds

z = number of phrases

t = number of corrections

$$z\log z + t\lceil\log(2\varepsilon+1)\rceil + \mathcal{O}\left(z\log\frac{u}{z}\right) \text{ bits}$$

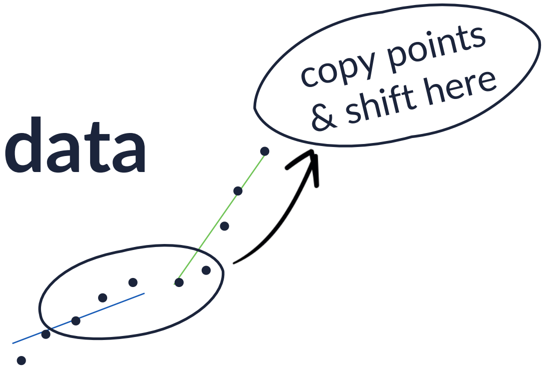
Term further bounded by $nH_k(\text{gap string}) + o\left(n\log\frac{u}{n}\right)$

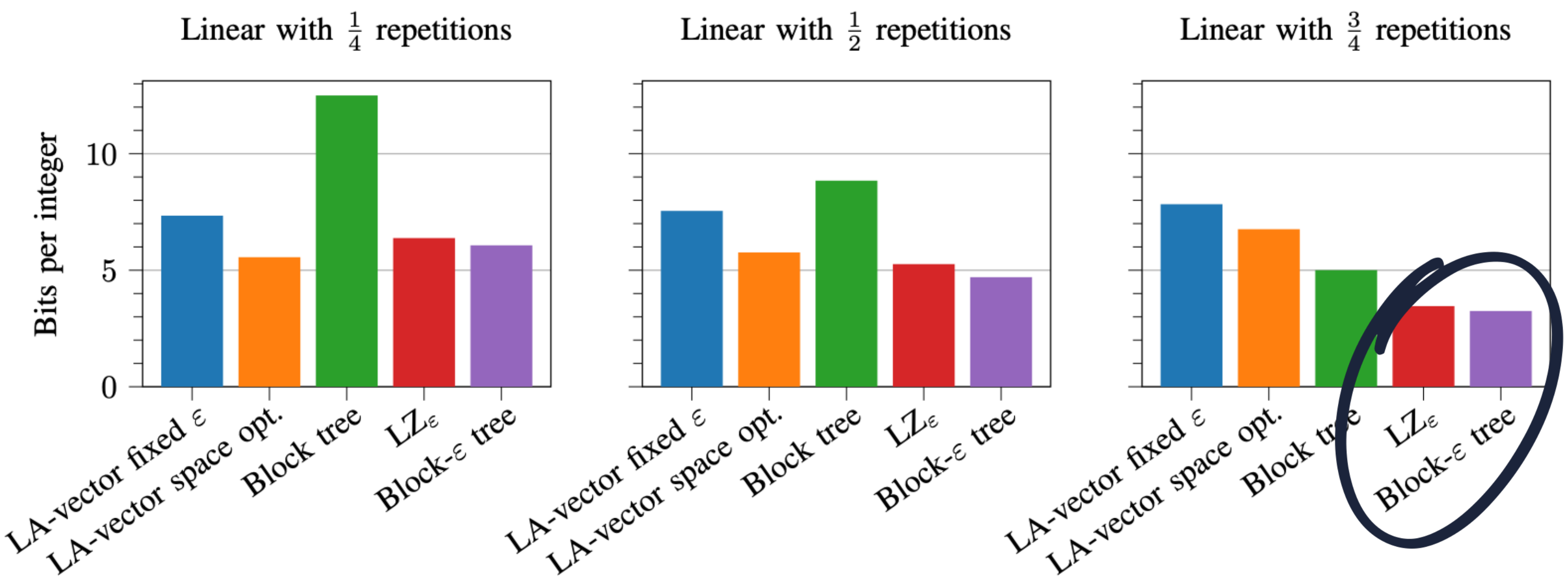
New results (submitted to journal)

3. Implemented and experimented the LZ_ε parsing on **real data**
   - 15× slower in rank than LA-vector, 5× slower in rank than block-ε tree
   - Worse in space than either LA-vector or block-ε tree…
   - … mainly because LZ_ε does not adapt to the “degrees of linearity” of the data by varying ε

New results (submitted to journal)

4. Experimented the LZ_ε and block-ε tree on **synthetic data**
   - Generate noisy data around random segments
   - With probability *p* = ¼, ½, ¾ create a copy





Both our proposals can take advantage of repetitions and linearity

Joint work with Paolo and Giovanni