RANDOM SAMPLING:

- given a sequence of items $S = (i_1, i_2, \dots, i_n)$ (→ $\mathbb{N}, \mathbb{R}$, strings ...) and a positive integer $m \leq n$ ⇒ select a subset of m items from S distributed (at random) (1) (uniformly) (2)

- n is known (or not) (if streaming ⇒ can't set prob in advance) if n is known ⇒ know prob. of extraction in advance ⇒ $P(\text{sampling an item}) = \frac{1}{n}$
- 1st DISK
- items occupy $\frac{n}{B}$ pages

- procedure RAND (A, B) : selects at random a number in range $[a, b]$
- want position of sampled items in SORTED ORDER: [2 CASES]: n known & n unknown

⇒ 1) speeds up the extraction from S (Disk & Stream)

2) Reduces working space

- extracts items efficiently via SCAN (no auxiliary array of pointers)

if $m > M$ ⇒ need a disk-based sorter

if $m \leq M$ ⇒ positions are integers in a fixed range ⇒ radix sort of items

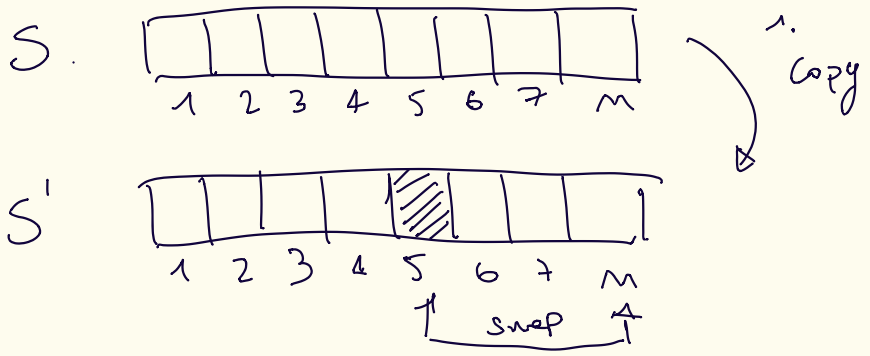
(1) n is known and 2 LEVEL MEMORY MODEL

DISK MODEL and KNOWN sequence length

- input size n is known
- $S[1, n]$ stored in a file on disk and cannot be modified

ALGORITHM

1. initialize the auxiliary array $S'[1, m] = S[1, m]$; // copy S in S' (array of pointers (we items may have variable length, es. strings))
2. for $s = 0, 1, \dots, m-1$ do
3. $p = \text{RAND}(1, n-s)$;
4. select the item (pointed by) $S'[p]$;
5. swap $S'[p]$ with $S'[n-s]$ ▷ s : # ultimi elementi
6. end for



2. $s := 0$ ⇒ # of sampled items

3. $p = rand(1, n-s)$ = item to be sampled ⇒ es. $p = RAND(1,8) = 5$

4: pick $S'[5]$

5. swap $S'[5]$ with last item $S'[n-s]$: $S'[8-0]$

Repeat the process:

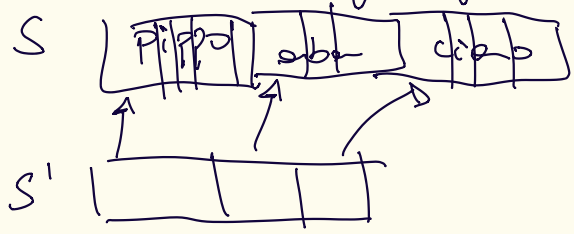
$s = 1$ => sample between 1 and $M - s = n - 1$

GOOD
- # of RANDOM numbers is M
- times good

BAD
- ADDITIONAL space (I'm copying the array
- LOT of I/Os, I'm jumping

remote good

If in S I've arbitrary length strings



es: URLs

=> In S' I can have pointers

=> S' is proportional to the # of objects, not total size of the set I wanna sample

$|S'| = n$ = # items (not to their length)

S = [31, 37, 53, 47, 97, 89, 11, 2, 3, 5]
0 1 2 3 4 5 6 7 8 9 $m = 4$

S' = S;

for s = 0 to m-1 do
p = rand(0, m-s)
a = S'[p]
swap S'[p] with S'[m-s-1]

1ST RUN FOR s = 0 TO 4-1 = 3

p = RAND(0, 10-0) = 9 (m-s)
1ST ITEM = 5
a = S'[9]
SWAP S'[9] = 5 WITH S'[10-0-1] = S'[9] = 5

s = 1 TO 3

p = RAND(0, 10-1) = 4
a = S'[4] = 97
SWAP S'[p] with S'[10-1-1] [m-s-1]
S'[8] = 3

— 2 —

s = 2 TO 3
p = rand ...

✓

Solution 2 (still n is known, avoiding jumps on disks)

- access items via scan-based access

=> assume I can grab the index of the sampled items (es. 20, 7, 5, 12)

=> sort the indexes
=> bring them sorted
=> access from left to right => still $O(n)$ I/Os

but better cuz moving left to right, head of disk is faster than go back and forth

=> find set of indexes for we sample? → list, hash table ...

ALGORITHM (using a dictionary DS)

- AVOIDS ITEMS SWAPPING VIA AN AUXILIARY ARRAY

Algorithm 3.2 Dictionary of sampled positions

1: Initialize the dictionary $\mathcal{D} = \emptyset$;
2: **while** ($|\mathcal{D}| < m$) **do**
3: $p = \text{Rand}(1, n)$;
4: if $p \notin \mathcal{D}$ insert it;
5: **end while**

assuming $m \leq M$

=> $\mathcal{D}$ stays in Memory

I/O free

- I'm not touching the disk

SORT $\mathcal{D}$

extract $\mathcal{D}$'s items from S (disk) (left to right

cost:

1) UNSUCCESFULL (extract a position already in $\mathcal{D}$)
2) SUCCESFULL (" " " $\notin \mathcal{D}$)

- CAN ASSUME $m < n/2$

How many time unsuccesfull

$$\mathbb{P}(p \in \mathcal{D}) = \frac{\text{positive events}}{\text{total \# of events}} = \frac{|\mathcal{D}|}{n} < \frac{m}{n} \leq \frac{n/2}{n} = \frac{1}{2}$$

UNSUCCESFUL

total # of events = space from which I sample p

- we can assume that $m < \frac{1}{2} n$ cuz if $m > \frac{1}{2} n$ can turn upside down the problem: in $\mathcal{D}$ I put the position I don't want to sample (negative stuff) (complement)

• time complexity $O(m)^{time}$ and $O(m)$ additional space

FACT 3.1 Algorithm 3.2 *based on hashing with chaining requires $O(m)$ average time and takes $O(m)$ additional space to select uniformly at random m positions in $[1, n]$. The average depends both on the use of hashing and the cost of re-sampling. An additional sorting-cost is needed if we wish to extract the sampled items of S in a streaming-like fashion. In this case the overall sampling process takes $O(\min\{m, n/B\})$ I/Os.*

if use a binary-search TREE $O(m \log m)$ time

SOLUTION (3) avoid D (related to birthday paradox)

1. extract m numbers in $[1, n]$
   if they are all distinct → sort them
   else REPEAT

↳ checked by sorting

• STREAMING MODEL, n known

Algorithm 3.4 Scanning and selecting

```
1: s = 0;
2: for (j = 1; (j ≤ n) && (s < m); j++) do
3:     p = Rand(0, 1);
4:     if (p < (m−s)/(n−j+1)) then
5:         select S[j];
6:         s++;
7:     end if
8: end for
```

S = (1, 12, 14)
    0 1 2
    s

WANT 2 samples : m

s = 0

for (j = 0 ; j < n && s < m ; j++)

p = rand (0,1);

if (p < (m − s)/(n − j + 1))

select S[j]

s++

end if

end for

s = 0 m

j = 0; 0 < 3 && 0 < 2

p = 0,4

if 0,4 < (2 − 0)/(2 − 0 + 1) = 2/3 = 0,6 yes

select S[0]

s = 1 m = 2

j = 1 ; 1 < 3 ; 1 < 2

p = 0,5 < (2 − 1)/(3 − 1 + 1) = 1/3 = 0,3 no 2 NOT picked

SOLUTION (h)

- n is known
- I'm considering STREAMING MODEL

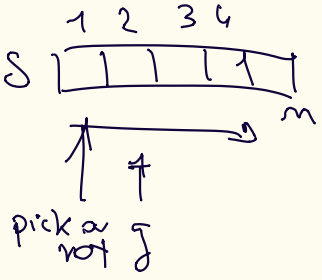
=> means S is looked from left to right

(no Dictionary of position, may have not much space in Memory, or search engines)

=> look item 1 (decide pick or not)

go forward

2 (pick or not)



=> NETWORK SITUATION: buffer only items I really pick

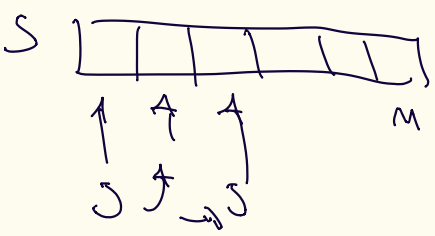
Algorithm 3.4 Scanning and selecting

```
1: s = 0;  # of sampled items
2: for (j = 1; (j ≤ n) && (s < m); j++) do
3:     p = Rand(0, 1);
4:     if (p < (m−s)/(n−j+1)) then
5:         select S[j];
6:         s++;
7:     end if
8: end for
```

guarantee I'm sampling m items

Real items che ho estratto finora

environment



$p = (0,1)$

I'm extracting n random numbers one for every Item in S I'm extracting a random number, according to read number, I decide to pick or not pick

case:
- $m = 1$ want to extract only 1 item

s starts from 0; as soon I've 1 item prob is $\frac{0}{\dots} = 0$

if $\left(p \geq \frac{m-s}{n-j+1}\right)$ ... PROB: $\frac{1-s}{n-j+1}$

S: $i_1 \; i_2 \; i_3 \; i_n \dots$

↑

a) $s = 0$

① $j = 1$
$P \leq \frac{1}{n}$

② $p = \text{rand}(0,1)$

if $p < \frac{1-0}{n-1+1}$ => here prob is $\leq \frac{1}{n}$

1st item sampled with prob $\frac{1}{n}$

proof by induction (on the index j)

case not sampled i_1, go to i_2

$i_1 \; i_2 \; i_3 \; i_4 \; \dots \; i_n$

↑ $j=1$ → ↑ $j=2$ $j=3$ and so on

$P \leq \frac{1}{n}$ $P \leq \frac{1}{n-1}$ $P = \frac{1}{n-3+1} = \frac{1}{n-2}$

s still 0 (no samples)

$P \leq \frac{1-0}{n-2+1}$

not $\frac{1}{n}$ but is missing that if I'm arriving here => means I failed sampling i_1 (before)

and so on

if I fail continuously and arrive to i_n

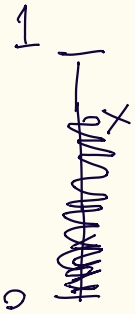
$j = n$

$P \leq \frac{1-0}{n-n+1} = \frac{1}{1}$ i_n is TAKEN

=> for sure I pick 1 item

.so: I'm taking the interval 0,1 , want guarantee that ALGORITH takes a choice according to some given probability (our case is $p < \frac{m-s}{n-j+1}$)

x is a prob $0 < x < 1$



$$Prob(p \leq x) = x$$

- So saying picking p uniformally ($p = rand(0,1)$)
- and compare p with x ($p < \frac{m-s}{n-j+1}$)

is equivalent to say:

this event: $p < \frac{m-s}{n-j+1}$ occurs with probability $\frac{m-s}{n-j+1} = x$

es: if I want to execute an event with probability $\frac{2}{3}$ and I pick a random number,

$\left(p \leq \frac{2}{3}\right) \longrightarrow x$

get p ; compare p with $\frac{2}{3}$

if it's true => do the event

if not => not do the event

=> P (I execute my event) = Prob my random number p falls here

but it's a uniform number

so IT'S $\frac{2}{3}$ / total size = 1

" $\frac{2}{3}$

Prove prob select 1 item is $\frac{1}{n}$

base case: Prob (to select the 1st item is $P \leq \frac{1}{n}$) $= \frac{1}{n}$

Compute Prob (pick i_j) (base case)

by inductive hp: assume → any item before item i_j is $\frac{1}{n}$

$\mathbb{P}$ (pick i_x, for $x < j$) $= \frac{1}{n}$

$\mathbb{P}$ (pick i_j) $=$ ($\mathbb{P}$ (I don't pick any items before i_j)) $\cdot$

$= \mathbb{P}(\neg \text{pick } i_1, i_2, \dots, i_{j-1}) \cdot \mathbb{P}(\text{pick } i_j)$

$1 - \left(\underbrace{P(\text{pick } i_1)}_{\frac{1}{n}} + \underbrace{P(\text{pick } i_2 \dots)}_{\frac{1}{n}} \right) =$ given by algo $= \frac{1}{n-j+1}$

$\frac{1-s=0}{n-j+1}$

$= \left(1 - \frac{j-1}{n}\right)$ ← every item is picked with prob $\frac{1}{n}$

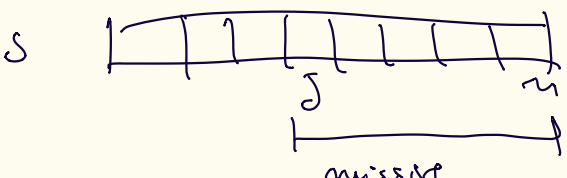
$\Rightarrow \left(1 - \frac{j-1}{n}\right) \cdot \left(\frac{1}{n-j+1}\right) =$

$= \left(\frac{n-j+1}{n}\right) \cdot \left(\frac{1}{n-j+1}\right) = \frac{1}{n}$

Generalize it to m (no pick 1 item but more)

$\mathbb{P}$ (pick i_j given that we have to pick others s elements)

(general situa in the for loop) =



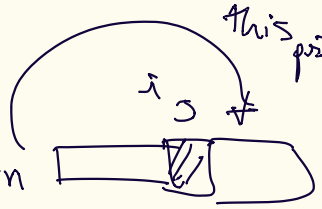
→ here have to pick $m - s$ ← # item already picked

How many ways I can pick $m-s$ items among $n-j+1$ items?

(total # of events (items) $= n-j+1$ choose $m-s$) = ~~total~~ possibility

$$\binom{n-j+1}{m-s}$$

$$P = \frac{\text{positive events} = i_j \text{ is picked}}{\binom{n-j+1}{m-s}} = \frac{\text{count \# of configurations } i_j \text{ is picked}}{\binom{n-j+1}{m-s}} =$$



picked i_j, among $m-s-1$, that can be choosed in

$$= \frac{\binom{n-j}{m-s-1}}{\binom{n-j+1}{m-s}} = \frac{m-s}{n-j+1}$$

another proof (easy): Lisa notes lesson 7

- Given the sequence $S = (a, b, c, d, e, f)$ $\quad n=6,\ m=2$

$$p = \left(\frac{1}{2}, \frac{1}{10}, \frac{3}{4}, \frac{3}{4}, 0, 1\right)$$

$\Rightarrow$ extract a random number p and compare it with $p < \frac{m-s}{n-j+1}$

- $s=0,\ j=1$

$$\frac{1}{2} \overset{?}{<} \frac{2-0}{6-1+1} = \frac{2}{6} \overset{1}{3} \quad \text{NO} \Rightarrow a \text{ is not picked}$$

- $s=0,\ j=2$ $\quad \frac{1}{10} \overset{?}{<} \frac{2-0}{6-2+1} = \frac{2}{5}$ $\quad$ Yes $\Rightarrow$ pick b

- $s=1$ $\quad j=3$ $\quad \frac{3}{4} \overset{?}{<} \frac{2-1}{6-3+1} = \frac{1}{4}$ $\quad$ NO

- $S=1 \quad j=4 \quad \frac{3}{4} \overset{?}{\leq} \frac{2-1}{6-4+1} = \frac{1}{3}$ NO

$S=1 \quad j=4 \quad 0 \overset{?}{\leq} \frac{1}{6-5+1} = \frac{1}{2}$ yes pick e

$S=2=6$, e => STOP

---

② N IS UNKNOWN, STREAMING MODEL

RESERVOIR SAMPLING

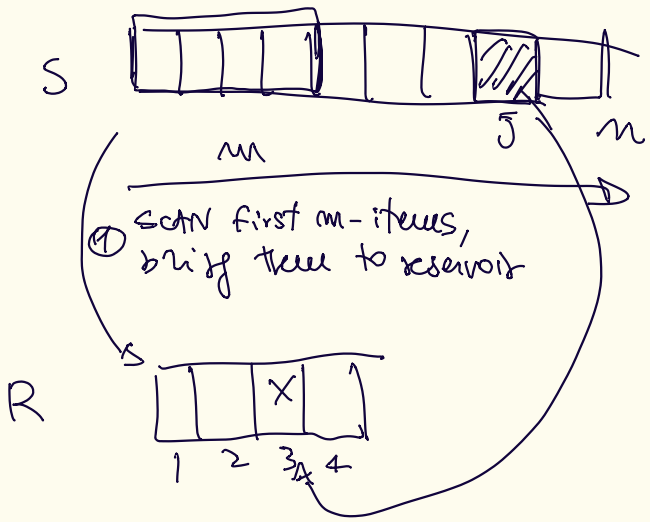
sampling with prob = $\frac{m}{n}$

Algorithm 3.6 Reservoir sampling

1: Initialize array $R[1,m] = S[1,m]$; # take first m items of the sequence and store in reservoir
2: **for** each next item $S[j]$ **do** //SCAN, where $j = m+1, \dots$ => $j > m$
3: $h = \text{Rand}(1, j)$; // 1 integer
4: **if** $h \leq m$ **then**
5: set $R[h] = S[j]$;
6: **end if**
7: **end for**
8: **return** array R;

(*)

$m=4$



② in any position j, extract an integer(h) from 1 to j
if the integer $\leq m$
$h \quad \leq \quad 4$
3
=> substitute the item in pos 3 of reservoir and put S[j]

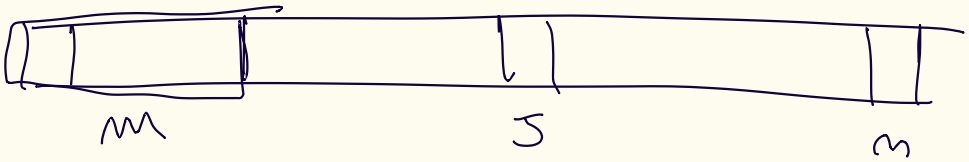
$P(j \text{ is stored in } R) = \frac{m}{j}$ $(h \leq m)$

when arrive to pos. n
$P(n \text{ goes in Reservoir}) = \frac{m}{n}$

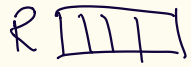
but h is taken from 1, j
- tot # of possibility = j
- possibility that are successful are m

notice that $j > m$ $(\text{so } P < 1)$

- when I'm at the end, the probability that an item is in the reservoir is $\frac{m}{n}$



$\frac{m}{n}$



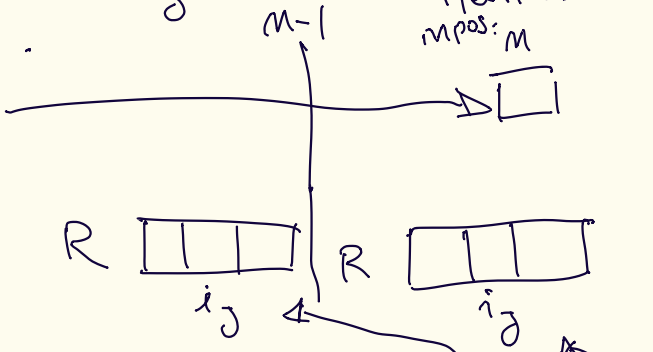
=> for sure last item is in Reservoir with $P = \frac{m}{n}$, but items before?

by induction on n: base case $n = m$ $P = \frac{m}{n} = 1$

(*) i_j is stored in R with probability $P = \frac{m}{j}$

WANT TO STAY

$P(\text{item } i_j \in R \text{ after } n \text{ steps}) =$ (i_j at pos m)



i_j to be in the reservoir at the last step, must be in R at pos n-1

$= P(i_j \in R \text{ after } n-1 \text{ steps}) \cdot \frac{m}{n}$ ← $m/n-1$

$[P(i_n \text{ is not taken}) + P(i_n \text{ is taken}) \cdot P(i_j \text{ is not picked out})] =$

$1 - \frac{m}{n}$ ← $\frac{m}{n}$

$\leq \frac{m-1}{n} = 1 - \frac{1}{n}$

i_j to stay in R, two possibilities: either i_n is not taken, R not change + OR i_n is taken • AND i_n is not written in position of i_j

$= \frac{m}{n-1} \cdot \left[\left(1 - \frac{m}{n}\right) + \frac{m}{n} \cdot \frac{m-1}{m} \right] =$

$= \frac{m}{n-1} \cdot \left[\frac{n-m}{n} + \frac{m-1}{n} \right] =$

$= \frac{m}{n-1} \cdot \left[\frac{n - m + m - 1}{n} \right] = \frac{m}{m-1} \cdot \frac{n-1}{n} = \frac{m}{n}$ ✓

N is known, streaming model, $m=2$ $n=8$

$S: (a, b, c, d, e, f, g, h)$

$P = [0,5 \mid 0,5 \mid 0 \mid 0,5 \mid 1 \mid 0 \mid 1 \mid 1] \leftarrow RAND(0,1)$

```
s=0
for ( j=1; (j<=n) && (s<m); j++)
        p = rand(0,1)
        if (p < (m-s)/(n-j+1))
        pick S[j];
end if  s++;
end for
```

$\Rightarrow s=0 \quad j=1$

$0,5 \overset{?}{<} \frac{m-s}{n-j+1} = \frac{2}{8-1+1} = \frac{2}{8} = \frac{1}{4} \Rightarrow$ no

$s=0 \quad j=2$

$0,5 \overset{?}{<} \frac{2}{8-2+1} < \frac{2}{7} \Rightarrow$ no

$s=0 \quad j=3$

$0 \overset{?}{<} \frac{2}{8-3+1} = \frac{2}{6} = \frac{1}{3} \Rightarrow$ yes $\Rightarrow$ pick c

$s=1 \quad j=4$

$\frac{1}{2} = 0,5 \overset{?}{<} \frac{1}{8-4+1} = \frac{1}{5} \Rightarrow$ no

$s=1 \quad j=5$

$1 \overset{?}{<} \frac{1}{8-5+1} = \frac{1}{4}$ no

$s=1 \quad j=6$

$0 \overset{?}{<} \frac{1}{8-6+1} = \frac{1}{3}$ yes $\Rightarrow$ pick f

STOP

RESERVOIR SAMPLING: (streaming, n is unknown)

Initialize $R[1,m] = S[1,m]$

for next item $S[j]$ // $j > m$

$h = rand(1, j)$ // h is an integer

if $(h \leq m)$

$R[h] = R[j]$

end if

end for

return R

es: WHAT IS THE STATUS of the Reservoir as soon we reach the item i?

$S = [a, b, c, d, e, f, g, h, i, \ldots]$ (4) (5)

1 2 3 4 5 6 7 8 9

$m = 3$

2 4 1 2 3 1 : (h) draw from $[1, j]$

current pos of the stream

$R =$

1	2	3
a	b	c

- consider char (d), is extracted in position 2 $(h \leq m)^{2 \leq 3}$ yes $\Rightarrow R[2] = S[4]$

$R =$

1	2	3
a	d	c

- consider (e) $(h \leq m)^{4 ? 3}$ NO => NIBBO

$R =$

1	2	3
a	d	c

- consider (f) $(h \leq m)^{?}$ $1 \leq 3$ yes $\Rightarrow R[1] = R[6]$

$R =$

1	2	3
f	d	c

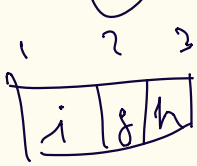
- consider (g) $(h \leq m)^{?}$ $2 \leq 3$ yes $\Rightarrow R[2] = R[7]$

$R =$

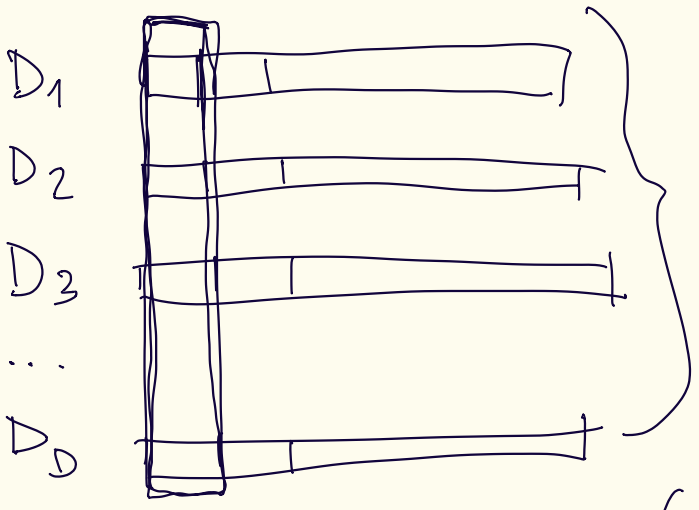
1	2	3
f	g	c

- consider (h) $(h \overset{?}{\leq} m)$ $(3 \leq 3)$ yes => swap R[3] = R[8]

- consider (i) $(h \overset{?}{\leq} m)$ yes => swap R[2] = R[5]



DISK STRIPING : D-disks :=> (D > 1)



- Instead of considering the page of the disk individual, consider "n" pages of disks all together

1 disk, where disk page B' is by

$$B' = D \cdot B$$

- In case of sorting, bound was $O\left(\frac{n}{B'} \log_{\frac{M}{B'}} \frac{n}{M}\right)$ over D-disk, with disk striping

(B' = D·B, $\frac{M}{B'}$ = $\frac{M}{DB}$)

lower bound, same but in $\log_{\frac{M}{BD}}$ less $\log \frac{n}{1B}$

- lower bound $\Omega\left(\frac{n}{DB} \cdot \log_{\frac{M}{B}} \frac{n}{M}\right)$

- How much is the slow down? I.E.: how much the disk striping losses against sorting lower bound?

RATIO: $\frac{O}{\Omega}$

$$\frac{\frac{N}{DB}\cdot \log_{\frac{M}{DB}} \frac{N}{M}}{\frac{N}{DB}\cdot \log_{\frac{M}{B}} \frac{N}{M}} \quad (\geq 1) \Rightarrow =$$

w.r. Disk striping is < than optimal

$$\frac{\log_2 \frac{N}{M}}{\log_2 \frac{M}{DB}} \cdot \frac{\log_2 \frac{N}{B}}{\log_2 \frac{N}{M}} = \frac{\log_2 \frac{N}{B}}{\log_2 \frac{M}{B} - \log_2 D} = \frac{\log_2 \frac{N}{B} : \log_2 \frac{M}{B}}{\log_2 \frac{N}{B} - \log_2 D : \log_2 \frac{M}{B}} =$$

$$= \frac{1}{1 - \frac{\log_2 D}{\log_2 \frac{N}{B}}} = \frac{1}{1 - \log_{\frac{N}{B}} D} \quad < 1 \Rightarrow \text{RATIO OVERALL} > 1$$

the base is of thousands and disk is tens, so

$$\Rightarrow \log_{\frac{N}{B}} D \ll 1$$

$N \rightsquigarrow +\infty \Rightarrow 1$ OPTIMAL

$\Rightarrow$ larger is the memory N, larger is the base, so the log goes to 0 $\Rightarrow$ ratio goes to 1 $\Rightarrow$ optimality

if N goes to $DB \Rightarrow$ ratio goes to $\infty \Rightarrow$ slow respect to optimal algorithm

Random sampling

```
s:=0
for ( j:=1 ; j<=n && s<m ; j++)
        p = rand(0,1)   => given by the exercise
        if ( p <= (m-s)/(n-j+1) )
            select S[j];
            s++;
        endif end for
```

Random Sampling $p \overset{?}{\leq} \frac{m-s}{n-j+1}$

$m=2$ $n=8$ items $S = \{ a, b, c, d, e, f, g, h \}$

a	b	c	d	e	f	g	h
$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	1	0	1	1

$j=1$ $s=0$

$p = \frac{1}{2}$

$p \overset{?}{\leq} \frac{2-0}{8-1+1} = \frac{2}{8} = \frac{1}{4}$ NOT picked

$j=2$ $s=0$

$p = \frac{1}{2} \overset{?}{\leq} \frac{2-0}{8-2+1} = \frac{2}{7}$ NOT picked

$j=3$, $s=0$ $\quad p = 0 \overset{?}{\leq} \frac{2-0}{8-3+1} = \frac{2}{6} = \frac{1}{3}$ yes, C is picked

$j=4$; $s=1$ $\quad p = \frac{1}{2} \leq \frac{2-1}{8-4+1} = \frac{1}{5}$ NOT picked

$j=5$; $s=1$ $\quad p = 1 \leq \frac{2-1}{8-5+1} = \frac{1}{4}$ NOT picked

$j=6$; $s=1$ $\quad p = 0 \leq \frac{2-1}{8-6+1} = \frac{1}{3}$ yes, F is picked

$s=2$ STOP

RESERVOIR SAMPLING

$m = 2$ $S = \{ a, b, c, d, e, f \}$

$h(a) = 1$ $h(b) = 1$ $h(c) = 1$ $h(d) = 3$ $h(e) = 2$ $h(f) = 1$

$R =$ | a | b |

$h(c) = 1 \overset{?}{\leq} m = 2$ yes $\Rightarrow R[1]$ with $S[3] = c$

$R =$ | c | b |

$h(d) = 3 \overset{?}{\leq} m = 2$ NO, NISBA

(still $R =$ | c | b |)

$h(e) = 2 \overset{?}{\leq} m = 2$ yes $\Rightarrow R[2] = S[5]$

$R =$ | c | e |

$h(f) = 1 \overset{?}{\leq} m = 2$ yes $\Rightarrow R[1] = S[6]$

$R =$ | F | E | ✓

MULTI-KEY QS

pivot always first sky in R

R:= (BUS, BATH, ABACUS, AARGH, CAT)

i=1

p

(BUS)
BATH
ABACUS
AARGH
CAT

i:1 →

p ↓ ABACUS i=1 →
R< AARGH

(p) ↓ BUS
R= BATH i=2 →

R> (p) CAT ✓

R<:/↓
R= p ABACUS i=2 →
AARG
R>:/

(p) BATH i=2 → ✓
R<
R= BUS i=3 → ✓
R>

R< AARG i=2 → ✓
R= ABACUS i=3 → ✓
R>

Reservoir sampling $m = 2$

$S = ($ a b c d e f $)$

1 2 3 4 5 6

3 1 4 2

$R =$ | A | B |

1 2

$h(c) \overset{?}{\leq} m$ no

$h(D) \overset{?}{\leq} 2$ yes

$h(D) = 1$

$R[2] = S[4]$

$R =$ | D | B |

1 2

$h(E) \overset{?}{\leq} 2$ no

$h(F) \leq 2$ yes $R[2] = S[6]$

$R =$ | D | F |

1 2