Example 2

(*) Sketch: home assignment

Given a set of N strings, sample a random subset of size M, print sorted. (according to input)

- all set is available in memory (classic)
- every string seen just once (streaming)

obs: Given two integers N, M, where $M \le N$, sample a random subset of size M from [1, N], output sorted.

Tools: RandInt (I, J) : generates random integer in [I, J]
RandReal (0,1) : real in [0, 1)

(1) Generate M random integers RandInt (1, N), then sort
- Time : $O(M \lg_2 M)$ mergesort
- space : $O(M)$
- Cons : duplicates ? possibly generate less. (resample)
  no time optimal, because of "log-slowdown".

(1') Set a bit in an array ⟹ no sort, yes scan, but $O(N)$ space (probably next)

(2) Generate only M ints by a SCAN (no sort) and M space
→ useful in a streaming environment
→ Pick item i with $\mathbb{P} = \frac{\text{\#restanti da campionare}}{\text{\#items left}}$

$\mathbb{P}(\text{pick } i) = \frac{M - m}{N - i}$

- Time = $O(N)$ [may be more than $M \lg M$, but scan-based]
- space = M [already sorted, output immed. when select]

CONS if $M \ll N$ this is bad [think to select m edges out of M in a graph // MST external]

③ [redacted]

Deploy randomness, good "average case"

- [redacted] M bins of size N/M : integer i goes to $i \times \frac{M}{N}$
- bins implemented as lists.
- because ints are random, each list has average length 1 (hint: bucket sort)

- Time: $O(M)$ average (even insertion sort is ok on each bin)
- Space: M

CONS: resampling, average complexity → { m extractions have cost of a conflict $\leq \frac{1}{N} + \frac{2}{N} + \dots + \frac{m-1}{N} \leq \frac{m^2}{N}$

④ Generate "gaps" rather than "indicators".

$G([redacted]^m, [redacted]^n)$ ⇒ random var that indicates the gap of the next element, when m have to be still selected out of n left.

$$\mathbb{P}(G = g) = f(g) = \binom{n-g-1}{m-1} \Big/ \binom{n}{m}$$

- the key problem is how to generate g according to f().
- Vitter proposed a sophisticated solution for this
  - paper attached
  - { linear time (on average)
    { no conflicts and [redacted] this no resampling

SEQUENCE TO SHOW

	Time	Space	Features
1'	N	N	scan only (tapes), no sort
2	N	M	Knuth (streaming) NO-RESAMPLE
1	M log M	M	sort-based + resample
3	M	M	average + resample Vitter ④ + stream