Derandomizing with PRG

- Indyk '00 showed a black-box way to decrease memory for such a construction using PRGs for small space algorithms

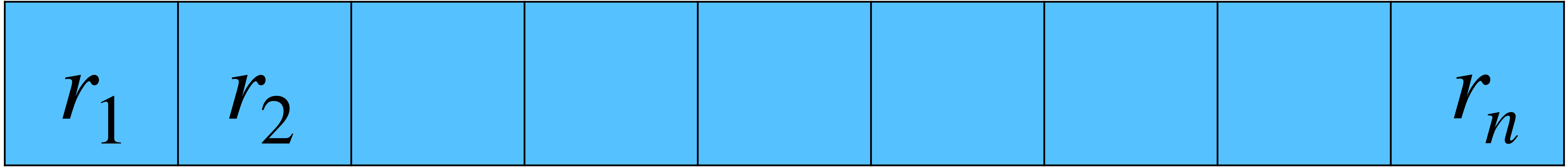
- Output distribution changes by a small amount

- Consider the **one-pass** algorithm parameterized by x which takes as input a random string with n blocks

- Initialize sk $\leftarrow 0$

- Use r_i to generate S_{*i}

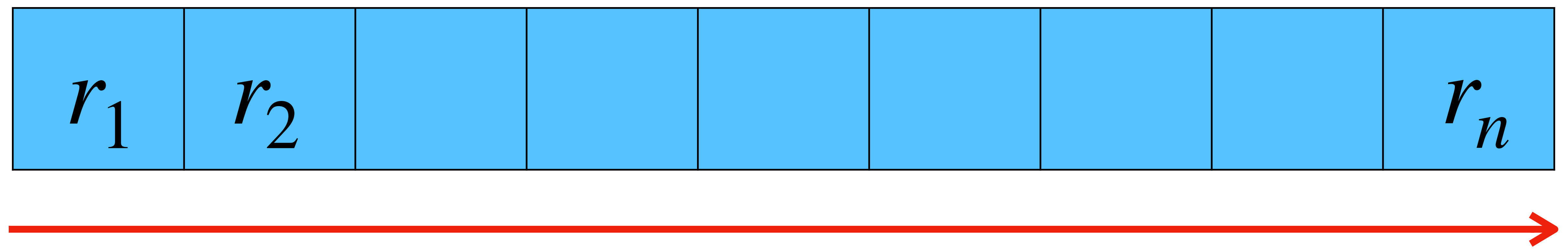
- Update sk $\leftarrow$ sk $+ x_i \cdot S_{*i}$


r_1
r_2
r_n

17

Derandomizing with PRG

- Indyk '00 showed a black-box way to decrease memory for such a construction using PRGs for small space algorithms
  - Output distribution changes by a small amount
- Consider the **one-pass** algorithm parameterized by x which takes as input a random string with n blocks
  - Initialize sk $\leftarrow 0$
  - Use r_i to generate S_{*i}
  - Update sk $\leftarrow$ sk $+ x_i \cdot S_{*i}$



Nisan's PRG