Selection in Column Monotone Matrices, X + Y and Heaps

[G.N. Frederickson, D.B. Johnson, *The Complexity of Selection and Ranking in X+Y and Matrices with Sorted Columns*, Journal of Computer and System Sciences 24(2): 197-208, 1982]

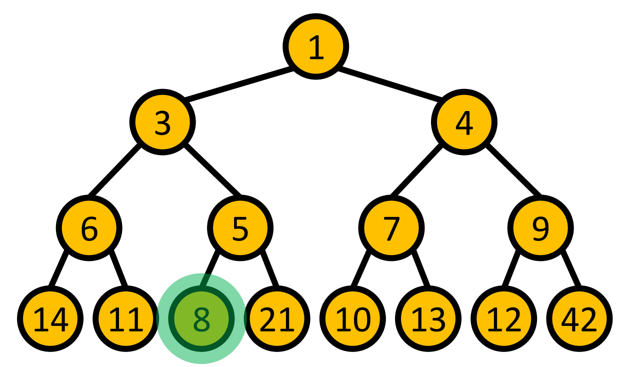
[G.N. Frederickson, *An Optimal Algorithm for Selection in a Min-Heap*, Inf. Comput. 104(2): 197-214, 1993]

	1	2	3		…		*m*
1	3	7	2	1	10	5	3
2	4	8	4	2	11	6	4
3	6	9	6	3	12	8	5
	7	13	8	5	13	9	7
⋮	8	17	10	7	14	10	8
	10	19	11	11	15	11	9
n	24	31	12	13	16	23	17

Column monotone

		1	2	3		…		*m*
		2	4	5	6	1	3	7
1	8	10	12	13	14	9	11	15
2	4	6	8	9	10	5	7	11
3	2	4	6	7	8	3	5	9
	1	3	5	6	7	2	4	8
⋮	3	5	7	8	9	4	6	10
	6	8	10	11	12	7	9	13
n	5	7	9	10	11	6	8	12

X + *Y*



Heap

= **Select(7)**

Partition (I_1, i, I_2)

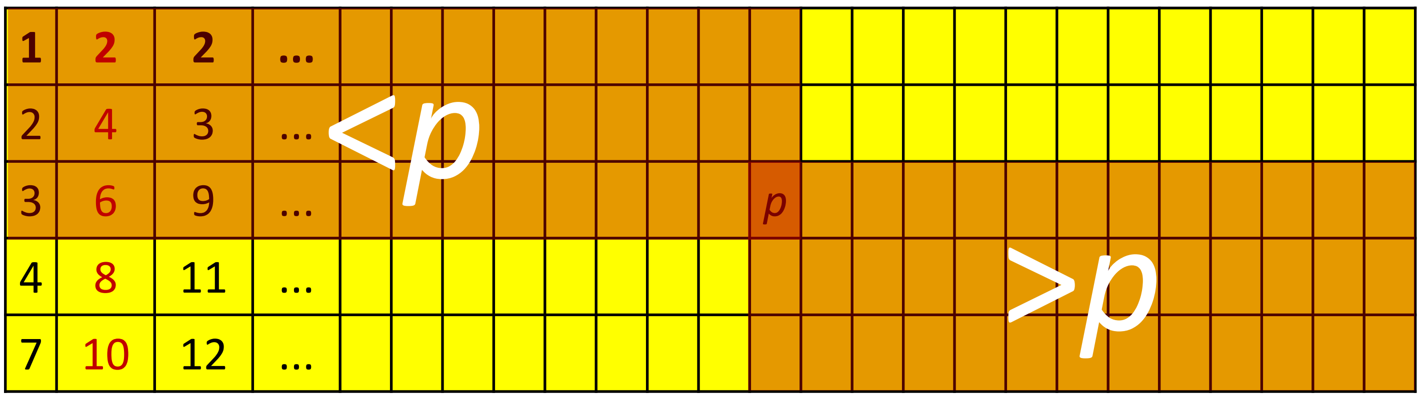
$j \in I_1 : x_j \le x_i \land j \in I_2 : x_i \le x_j$

i	1	2	3	4	5	6	**7**	8	9	10
x_i	10	15	7	33	42	17	**17**	11	17	7

Select(k) $\equiv$ find partition with $|I_1|+1 = k$

Select(6)

[M. Blum, R.W. Floyd, V. Pratt, R. Rivest and R. Tarjan, *Time bounds for selection*, J. Comp. Syst. Sci. 7 (1973) 448-461]



$T(n) = n + T(n/5) + T(7n/10) = O(n)$

Weighted-Select(w)

Find partition with $w - w_i \le \sum_{j \in I_1} w_j < w$

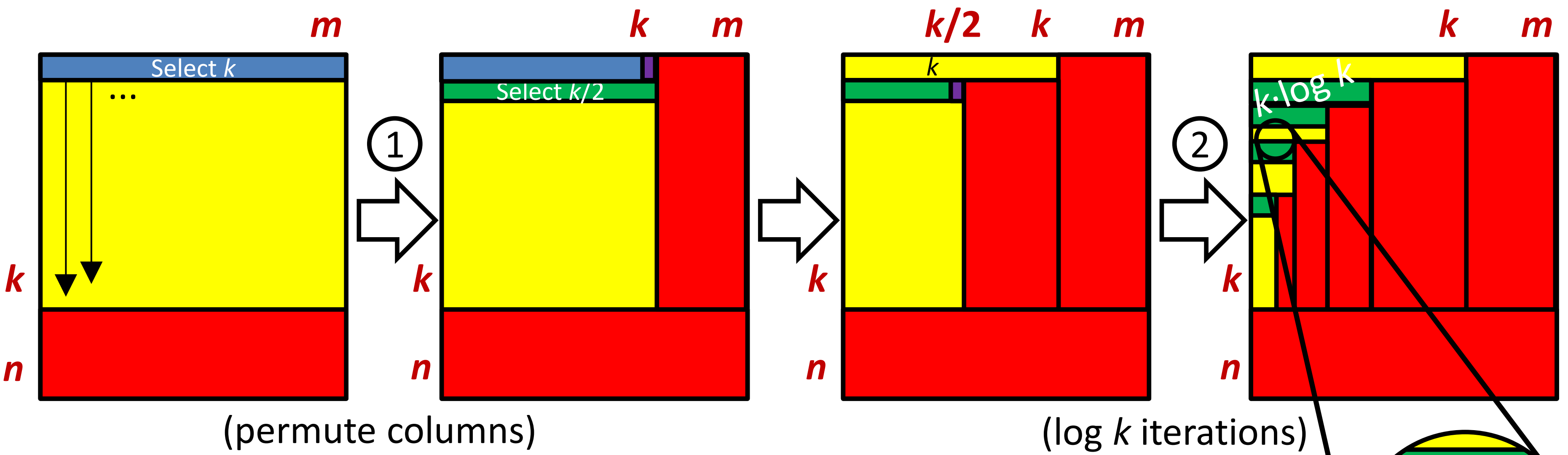
i	1	2	3	4	5	6	**7**	8	9	10
x_i	10	15	7	33	42	17	**17**	11	17	7
w_i	3	2	1	4	2	5	**7**	2	3	5

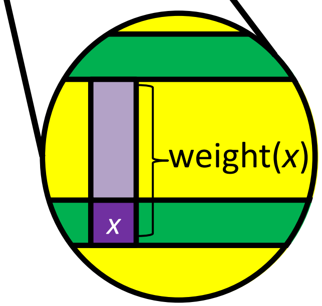
Weighted-Select(18)

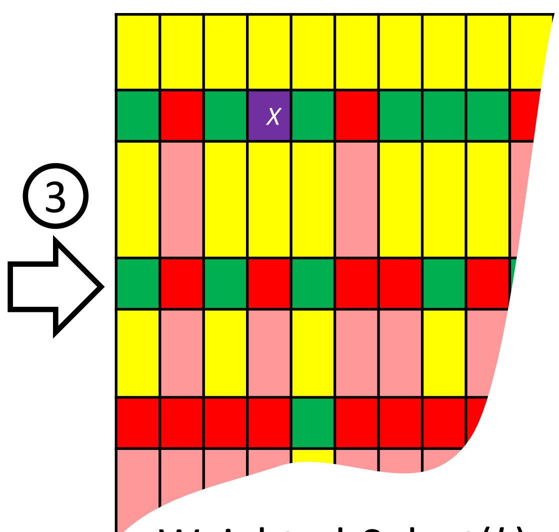
Algorithm : Binary search using Select

Time : $O(n + n/2 + n/4 + \cdots + 2 + 1) = O(n)$

Selection in Column Monotone Matrices







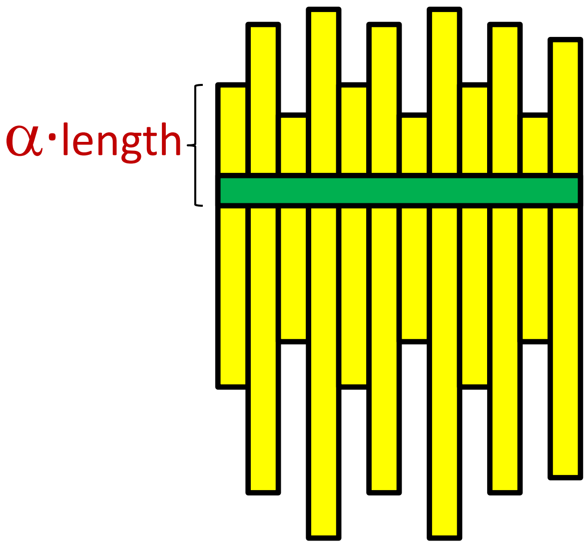
Weighted-Select(k)
on $k+k/2+k/4+\cdots+1$= O(k) elements
$2k$ candidates left (green + yellow)

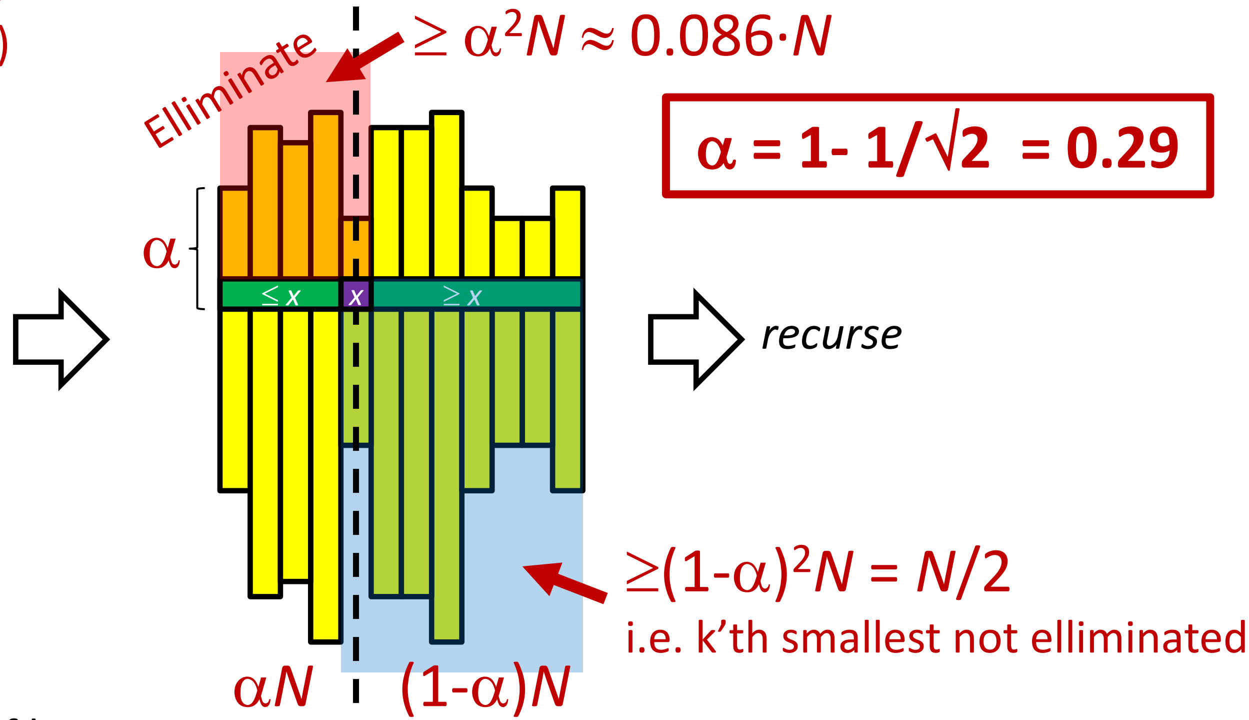
Result so far...
Identified O(k) elements in prefixes of p = min{m,k} columns

Time so far...
$O(m + p + p/2 + p/4 + \cdots + 1) = O(m)$

Selection in Column Monotone Matrices

Select(k,N,p) assume $k \geq N/2$
(N = total length of the p lists)



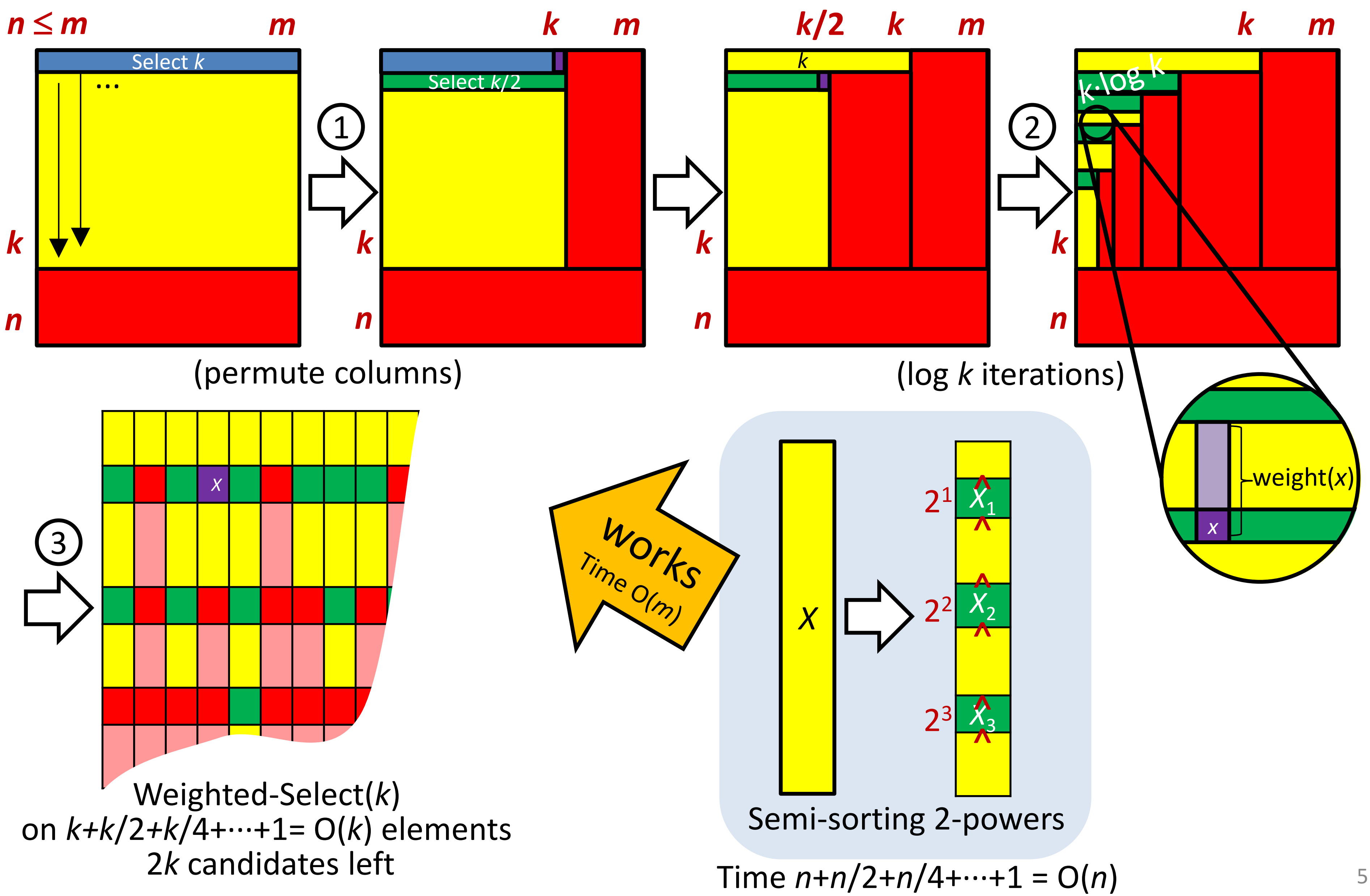


Weighted-Select(αN)
Weight of element = length of list

- N = O(p) $\Rightarrow$ Select(k)
- $k < N/2 \Rightarrow$ symmetric with reverse order
- $T(N) = p + T(\,(1-\alpha^2)\cdot N) = O(p \cdot \log\,(N/p))$

Total time O(m+p·log(k/p)), p = min { k, m}
$k = O(m) : O(m)$
$k = \Omega(m) : O(m \cdot \log(k/m))$

Selection in *X* + *Y* – reuse column monotone algorithm ?

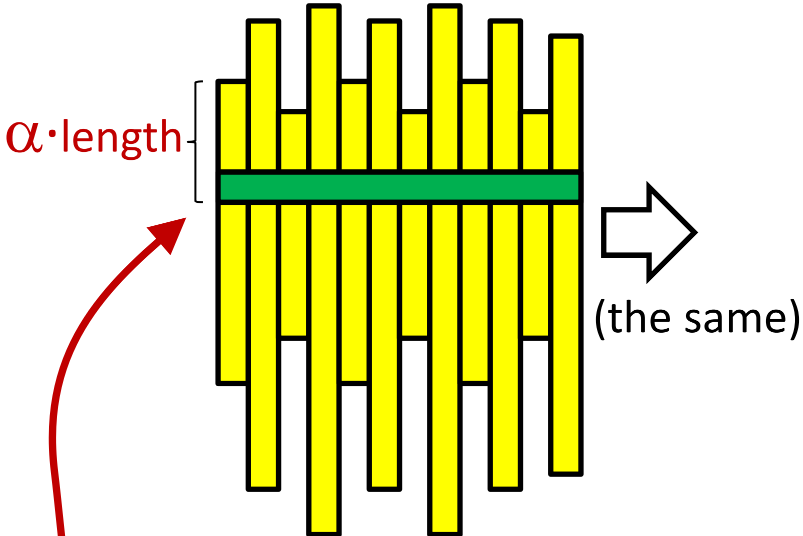


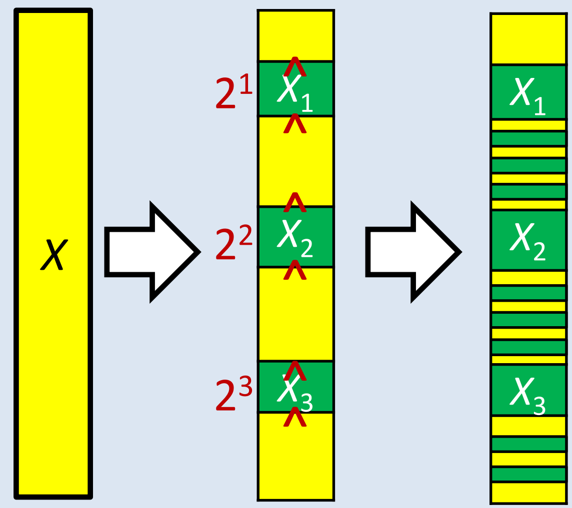
Selection in $X + Y$ – reuse column monotone algorithm ?

Select(k,N,p) assume $k \geq N/2$
(N = total length of the p lists)

$\alpha = 1/4$

Total time $O(m+p \cdot \log(k/p))$, $p = \min\{k, m\}$
$k = O(m)$: $O(m)$
$k = \Omega(m)$: $O(m+k \cdot \log(k/m))$





Each iteration refine semi-sorting
⇓
$O(n)$ per level

Semi-sorting 2-powers

Time $n+n/2+n/4+\cdots+1 = O(n)$

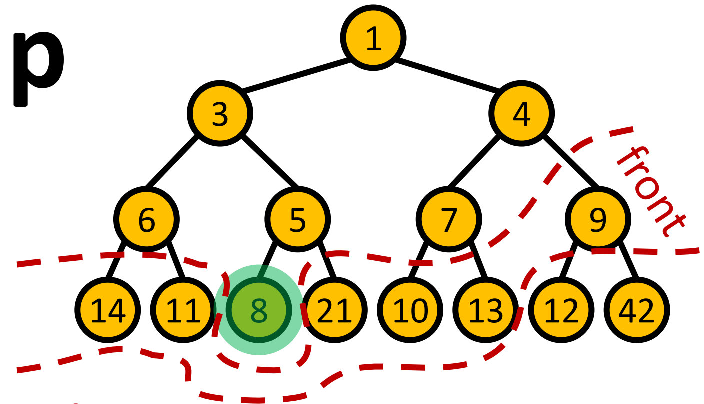
Can approximate sample by a close enough semi-sorted element

[F93] considers additionally...

- How to compute rank(x) in column monotone and X+Y matrices in **$O(m+p\cdot\log(k/p))$, p = min $\{k, m\}$**
- Proves that the bounds are optimal

Selection in a Binary Heap

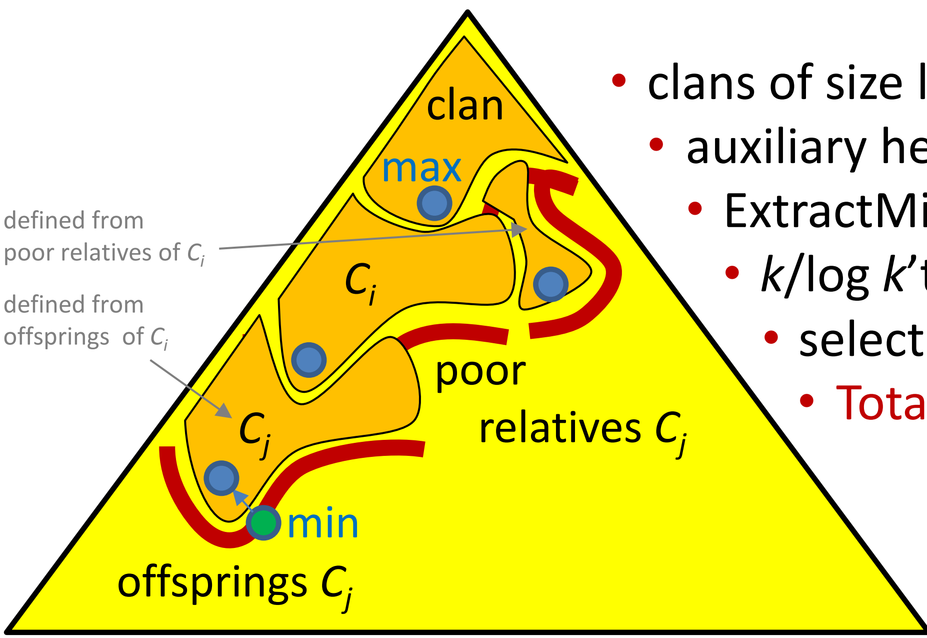
[G.N. Frederickson, *An Optimal Algorithm for Selection in a Min-Heap*, Inf. Comput. 104(2): 197-214, 1993]



- k x DeleteMin $\Rightarrow$ $O(k \cdot \log n)$
- k x DeleteMin **front** $\Rightarrow$ $O(k \cdot \log k)$

k smallest in sorted order $\Rightarrow$ $\Omega(k \cdot \log k)$ lower bound
only the kth smallest $\Rightarrow$ $\Omega(k)$ lower bound

$k \cdot \log k \rightarrow k \cdot \text{loglog } k \rightarrow k \cdot \text{logloglog } k \rightarrow k \cdot 3^{\log^* k} \rightarrow k \cdot 2^{\log^* k} \rightarrow k$



- clans of size $\log k$
  - auxiliary heap of clan representatives (max)
    - ExtractMin representative & construct 2 clans
      - $k/\log k$'th ExtractMin representative x has rank $k..2k$
        - select(k) on all elements $\leq x$
        - Total time $O(k \cdot \text{loglog } k + k/\log k \cdot \log k + k)$

clan construction — auxiliary heap — selection