

assume the matrix is  $n \times m$  and  $m \leq n$ .

1. Given a value  $x$ , we can know the rank of  $x$  in the matrix in  $O(n)$  time, by monotone pointers. Using binary search (on  $n^2$  elements), the running time is  $O(n \log U)$  or  $O(n \log n)$ .
  2.  $O(\min(k, n))$ . <https://chaoxuprime.com/posts/2014-04-02-selection-in-a-sorted-matrix.html>. Also  $O(n)$  [4].
  3. put the first element in each row into a heap, whenever we pop, add the next element in the corresponding row to the heap.  $O(k \log n)$ .
  4.  $O(\sqrt{k} \log k)$ . The crucial observation is that there are  $xy$  elements smaller than entry  $(x, y)$ , so the possible region for the solution can be divided into  $2\sqrt{k}$  rows and columns (i.e. row/columns  $1, \dots, \sqrt{k}$ ). use results on selection in union of sorted arrays.
- see reference for 4. median of two sorted arrays.  
optimal solution:  $\Theta(h \log \frac{2k}{h^2})$ , where  $h = \min\{\sqrt{k}, m\}$ . For  $k \leq m^2$ , this is  $\Omega(\sqrt{k})$ . see [2], and its erratum [3].

note. is [1] useful to get  $O(k)$ ?

## References

- [1] Greg N Frederickson. An optimal algorithm for selection in a min-heap. *Information and Computation*, 104(2):197–214, 1993.
- [2] Greg N Frederickson and Donald B Johnson. Generalized selection and ranking: sorted matrices. *SIAM Journal on computing*, 13(1):14–30, 1984.
- [3] Greg N Frederickson and Donald B Johnson. Erratum: generalized selection and ranking: sorted matrices. *SIAM Journal on Computing*, 19(1):205, 1990.
- [4] Andranik Mirzaian and Eshrat Arjomandi. Selection in  $x+y$  and matrices with sorted rows and columns. *Information processing letters*, 20(1):13–17, 1985.