

CPSC 320: TUTORIAL 11

1. We are given a matrix M of positive integers that has 4 rows and n columns. We want to circle entries of the matrix with maximum total sum. However, we cannot circle two entries that are next to each other in a row or column (diagonal adjacency is fine).

For example, in the following matrix, the circled entries are legal and have total value 88 but they do not have maximum total sum.

1	④	8	⑨	6	⑤	4	2	⑤	6
④	2	1	3	⑨	3	⑦	2	4	⑨
3	5	⑨	6	3	2	1	⑨	2	6
3	②	6	②	1	⑥	3	4	⑧	8

How many different patterns of circled entries can appear in one column of a legal choice?

Call two patterns *compatible* if they can be used in adjacent columns of a legal choice.

The solution to a subproblem consisting of the first i columns of M can be assigned a *type*, which is the pattern occurring in the last column.

Using the notions of compatibility and type, give a dynamic programming algorithm to compute an optimal choice in $O(n)$ time.