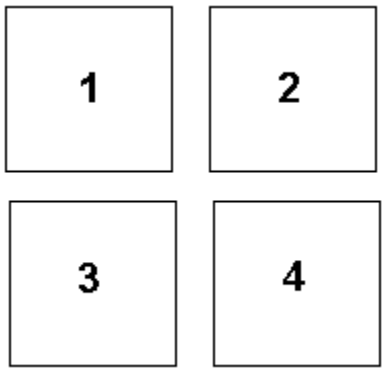
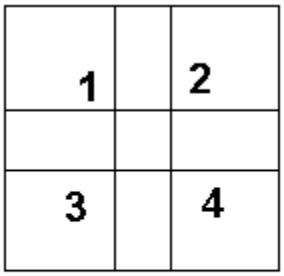
Poze – autor Negruseri Cosmin

Solutia 1 (suffix arrays 2d):
Se compara intre ele patratele de latura 2^k. Fiecare patrat va avea un cod format din patru numere. Cele patru numere sunt date de indecsii in ordinea sortata a celor patru subpatrate de latura 2^(k-1) care formeaza noul patrat. Pentru ca doua patrate sa fie egale trebuie ca si cele patru subpatrate care le compun sa fie la randul lor egale, deci cele patru coduri sa fie identice. Cat timp exista patrate cu coduri egale putem sa il marim pe k. Apoi pentru a determina toate cifrele in baza 2 a laturii maxime cautate sortam folosind radix sort patratele de orice latura L mai mare sau egala ca 2^k in lungime prin determinarea unui cod a patru patrate de latura 2^k care se suprapun si reuniunea lor formeaza un patrat de latura L. Aceasta solutie merge pe acelasi principiu pe care merge rezolvarea problemei mai simple a determinarii unei subsecvente a unui sir de caractere ce apare de cel putin k ori in sirul nostru (aplicatie clasica clasica a structurii de date suffix arrays). Ea are complexitatea O(n^2 log n), dar are constanta multiplicative destul de mare.





Solutia 2:
Se poate incerca implementarea unei cautari binare pe latura maxima. Pentru o latura fixata L vom determina codurile hash pentru fiecare submatrice de dimensiune LxL. Implementarea unui cod de hashing bun s-ar putea dovedi dificila, intai din punctul de vedere al datelor care merg pana la 30000 si astfel trebuie sa calculam pe 64 de biti, si in al doilea rand trebuie sa folosim o baza mare pentru a evita coliziunile. O solutie care scapa de unele din aceste probleme ar fi sa folosim mai multe functii de hashing in paralel si sa consideram doua matrici egale doar cand toate codurile hash asociate sunt egale. Aceasta solutie are complexitate O(n^2 log ^2 n) dar are constanta multiplicative destul de mica.