1. s jakou složitostí vzledem k R,S,K dokážete tuto úlohu řešit

Při pomocném předvýpočtu trvajícím pak budeme pro každý dotaz schopni odpovědět za konstatní počet operací, pro každou operaci tedy

1. jakým algoritmem byste ji řešili (můžete popsat slovy nebo napsat (pseudo)kód).

Definujme si pojem rohová matice jako matici, která má levý horní rohový prvek na souřadnících 1,1 a pravý dolní rohový prvek na souřadnicích

Provedením předvýpočtu lze dotazy řešit v konstantním čase, spočteme si počty jedniček ve všech rohových maticích.

Z takovou informací můžeme matici s hraničními souřadnicemi levého horního rohu a pravého dolního rohu vypočtítat jako počet jedniček jako počet jedniček v rohové matici , zmenšený o počty jedniček rohových maticích a a , zvětšený o počet jedniček v rohové matici , pro objasnění náčrt:



Pro získání počtu jedniček v černém obdelníku odečteme od počtu jedniček v zeleném vnějším obdelníku počty jedniček v modrých obdelnících a přičteme počet jedniček v oranžovém obdelníku.

Předvýpočet pro každou rohovou matici zabere operací, algoritmus lze popsat třeba takovýmto pseudokódem:

vstup: matice = int[rows][columns]

int[][] aux = int[row][column] //pomocné pole s počty jedniček pro rohové matice

foreach row:

int ones\_in\_row = 0

foreach column:

ones\_in\_row += matice[row][column]

if row == 0:

aux[row][column] = ones\_in\_row

else:

aux[row][column] = aux[row - 1][column] + ones\_in\_row

výstup: pomocné pole aux;

Každý dotaz pak vyhodnotíme jako v konstantním čase.

Nabízí se možnost optimalizace pro případy jako třeba když budeme vědět, že dostaneme jen jeden dotaz, pak je předvýpočet zbytečnou komplikací, nebo když každý dotaz bude na matici menší než je původní, pak stačí předvýpočet pro tuto zmenšenou matici, obecně je ale lepší výše popsané řešení.