Назив проблема: Скакач

|  |  |
| --- | --- |
| Аутор: Душко Обрадовић | Анализа: Никола Милосављевић |

|  |  |
| --- | --- |
| Тагови: | ad-hoc |

## Решење и анализа:

За комплетно решење задатка, довољно је испитати сва поља на које скакач може доћи у једном, односно у два потеза. За ово је најједноставније направити два константна низа: i ; тада скакач са поља може скочити на поља облика , за свако , која су унутар табле.

За , такође генеришемо све могуће (дво)потезе (њих 64) при чему проверавамо да ли је скакач унутар табле и после првог и после другог потеза. Приметимо да је могуће да смо на неко поље стигли на 2 начина али то поље треба рачунати само једном у решењу. Један од начина за избацивање дупликата је да у низу памтимо сва различита поља до којих смо до тада могли доћи у два потеза и, уколико се тренутно поље не налази у низу, додајемо га и повећавамо решење за 1. Следећи псеудокод илуструје ову идеју

================================================================

1. Niz obidjena\_polja = (); sol = 0;
2. for i = 1 to 8 do

03 if ((x+dx[i], y+dy[i]) unutar table)

04 for j = 1 to 8 do

05 if ((x+dx[i]+dx[j], y+dy[i]+dy[j]) unutar table i ne nalazi se

06 u nizu obidjenja\_polja) then

07 sol = sol + 1;

08 оbidjena\_polja[sol]=(x+dx[i]+dx[j], y+dy[i]+dy[j]);

================================================================

Рачунање неког поља 2 пута смо могли избећи и користећи чињеницу да скакач после 2 потеза може отићи највише 4 поља у сваком смеру – довољно је “одсећи” одговарајући део табле димензије не веће од и једноставно маркирати одговарајућа поља у добијеној малој матрици.

Временска и меморијска сложеност алгоритма је у сваком случају .