



## International Olympiad in Informatics 2012

23-30 September 2012

Sirmione - Montichiari, Italy

Competition tasks, day 1: Leonardo's inventions and projects

---

# Rrugematesi me guralece

---

Leonardo shpiku "rrugematesin" origjinal: nje karroce e cila mund te maste distancat duke hedhur guralece ndersa rrotat e karroces rrotullohen. Numerimi i guraleceve jep numrin e rrotullimeve te rrotes, duke i lejuar perdoruesit te llogarise distancen e pershkruar nga rrugematesi. Si shkencetare kompjuteri, i kemi shtuar nje kontroll softuerik rrugematesit, duke zgjeruar funksionalitetet e tij. Detyra juaj eshte te programoni rrugematesin sipas rregullave te specifikuara me poshte.

### Operacionet e grides

Rrugematesi leviz ne nje matrice imagjinare me katrore me permasa  $256 \times 256$  qeliza njesi. Cdo qelize mund te permbaje se shumti 15 guralece dhe identifikohet nga nje cift koordinatash (rresht, kollone), ku cdo koordinate eshte ne intervalin  $0, \dots, 255$ . Pasi eshte dhene nje qelize  $(i, j)$ , qeliza afer saj eshte (nese ekziston)  $(i-1, j)$ ,  $(i+1, j)$ ,  $(i, j-1)$  dhe  $(i, j+1)$ . Cdo qelize qe shtrihet ne qelizen e pare ose te fundit, ose ne kollonen e pare ose te fundit, quhet nje "kufi". Rrugematesi gjithmone fillon ne qelizen  $(0, 0)$  (kulmi veri-perindimor), duke u perballur me veriun.

### Komandat baze

Rrugematesi mund te programohet duke perdorur komandat e meposhtme.

- `majtas` — kthehu 90 grade majtas (ne drejtim te akrepave te ores) dhe qendro ne qelizen korrente (p.sh. nese perpara eshte perballe me jugun, atehere do te jete perballe me lindjen pas komandes).
- `djathtas` — kthehu 90 grade djathtas (ne drejtim te akrepave te ores) dhe qendro ne qelizen korrente (p.sh. nese perpara eshte perballe me perendimin, atehere do te jete perballe me veriun pas komandes).
- `leviz` — leviz nje njesi perpara (ne drejtimin perballe te cilit eshte rrugematesi) drejt qelezes fqinje. Nese nje qelize e tille nuk ekziston (p.sh kufijte ne ate drejtim jane arritur tashme) atehere kjo komande nuk ka efekt.
- `merr` — heq nje guralec nga qeliza korrente. Nese qeliza korrente nuk ka guralece, atehere komanda nuk ka efekt.
- `vendos` — shton nje guralec ne qelizen korrente. Nese qeliza korrente permban tashme 15 guralece, atehere komanda nuk ka efekt. Rrugematesi nuk zbraset asnjehere nga guralecet.
- `ndalo` — perfundon ekzekutimin.

Rrugematesi ekzekuton komandat ne rendin qe ato i jipen programit, nje komande per rresht. Rreshtat bosh injorohen. Simboli # tregon nje koment; cdo tekst qe ndiqet, deri ne fund te rreshtit, injorohet. Nese rrugematesi arrin fundin e programit, ekzekutimi perfundon.

### Shembull 1

Konsideroni programin e meposhtem per rrugematesin. Ai con rrugematesin ne qelizen (0,2) duke patur perballe lindjen. (Vini re qe komanda e pare `leviz` injorohet, sepse rrugematesi eshte ne qoshen veri-perendim duke patur perballe veriun.)

```
leviz # asnje efekt
djathtas
# tani rrugematesi ka perballe lindjen
leviz
leviz
```

### Etiketa, bordura dhe guralece

Per te ndryshuar rrjedhen e programit ne varesi te statusit korrent, ju mund te perdorni etiketat, te cilat jane sensitive nga dallimi i shkronjave te medha nga te voglat te perbera nga 128 simbole te zgjedhura nga a, ..., z, A, ..., Z, 0, ..., 9. Komandat e reja qe kane lidhje me etiketat jane listuar me poshte. Ne pershkrimin e meposhtem, *L* paraqet cdo etikete valide.

- `L:` (p.sh. `L` e ndjekur nga dy pika ':') — deklarone nje vendndodhje brenda programit me etikete *L*. Te gjitha etiketat e deklaruar duhet te jene unike. Deklarimi i nje etikete nuk ka efekt ne rrugematesin.
- `kerce L` — vazhdon ekzekutimin duke kercyer ne menyre te pakushtezuar drejt vizes me etiketen *L*.
- `border L` — vazhdon ekzekutimin duke kercyer drejt vizes me etiketen *L*, nese rrugematesi ne bordure ka perballe buzen e grides (p.sh. nje instruksion `leviz` nuk do te kishte asnje efekt); perndryshe, ekzekutimi do te vazhdoje ne menyre normale dhe kjo komande nuk do te kete asnje efekt.
- `guralec L` — vazhdon ekzekutimin duke kercyer drejt vizes me etikete *L*, nese qeliza korrente permban te pakten nje guralec; perndryshe, ekzekutimi vazhdon ne menyre normale dhe kjo komande nuk ka efekt.

### Shembull 2

Programi i meposhtem vendos guralecin e pare ne rreshtin 0 dhe ndalon atje; nese nuk ka asnje guralec ne rreshtin 0, ai ndalon ne bordure ne fund te rreshtit. Ai perdor dy etiketa `leonardo` dhe `davinci`.

```
djathtas
leonardo:
guralec davinci # guraleci u gjet
border davinci # fundi i rreshtit
```

```
leviz
kerce leonardo
davinci:
ndalo
```

Rrugematesi fillon duke u kthyer ne te djathten e tij. Cikli fillon me deklarimin e etiketes `leonardo`: dhe perfundon me komanden `kerce leonardo`. Ne cikel, rrugematesi kontrollon per prezencen e nje guraleci ne bordure dhe ne fund te rreshtit ; nese nuk eshte keshtu, rrugematesi ben `leviz` nga qeliza korrente  $(0, j)$  drejt qelizes fqinje  $(0, j + 1)$  persa kohe qe kjo ekziston. (Kodi `ndalo` nuk eshte domosdoshmerisht i nevojshem ketu qe programi te perfundoje)

## Detyra

Ju duhet ngarkoni nje program ne gjuhen e rrugematesit, te pershkruar si me siper, qe ben qe rrugematesi te sillet sic pritet. Cdo nendetyre (shiko me poshte) specifikon nje sjellje qe rrugematesit i kerkohet te permbushë dhe kufizimet qe qe zgjidhja e ngarkuar duhet te kenaqe. Kufizimet kane lidhje me dy ceshtjet e meposhtme.

- *Madhesia e programit* — programi duhet te jete I shkurter mjaftueshem. Madhesia e programit eshte numri i komandave ne te. Deklarimi i etiketave, komentet dhe rreshtat boshe nuk llogaritet ne madhesi.
- *Gjatesia e ekzekutimit* — programi duhet te perfundoje shpejt mjaftueshem. Gjatesia e ekzekutimit eshte numri i hapave te ekzekutuara: cdo komande ekzekutimi llogaritet sin je hap, pavaresisht nese komanda ka efekt apo jo, deklarimi I etiketave, komentet dhe rreshtat bosh nuk llogariten si hapa.

Ne shembullin 1, madhesia e programit eshte 4 dhe gjatesia e ekzekutimit eshte 4. Ne shembullin 2 madhesia e programit eshte 6 dhe, kur ekzekutohet ne nje gride me nje guralec te vetem ne qelizen  $(0, 10)$ , gjatesia e ekzekutimit eshte 43 hapa: `djathtas`, 10 iteracione te ciklit, cdo iteracion merr 4 hapa (guraleci `davinci; border davinci; leviz; kerce leonardo`), dhe me ne fund, guralec `davinci` dhe `ndalo`.

## Nendetyra 1 [9 pike]

Ne fillim Ka  $x$  guralece ne qelizen  $(0, 0)$  dhe  $y$  ne qelizen  $(0, 1)$ , ndersa te gjitha qelizat e tjera jane bosh. Kujtoni se mund te kete te shumten 15 guralece ne cdo qelize. Shkruani nje program qe perfundon me rrugematesin ne qelizen  $(0, 0)$  nese  $x \leq y$ , dhe ne qelizen  $(0, 1)$  perndryshe. (Nuk na intereson perballe cilit drejtim do te jete rrugematesi ne fund; neve nuk na intereson gjithashtu se sa guralece jane present ne fund te grides ose vendndodhjet e tyre.)

*Limitet:* madhesia e programit  $\leq 100$ , gjatesia e ekzekutimit  $\leq 1\,000$ .

## Nendetyra 2 [12 pike]

Enjehta detyre si me siper por kur programi te perfundoje, qeliza  $(0, 0)$  duhet te permbaje ekzaktesisht  $x$  guralece dhe qeliza  $(0, 1)$  duhet te permbaje ekzaktesisht  $y$  guralece.

*Limitet:* madhesia e programit  $\leq 200$ , gjatesia e ekzekutimit  $\leq 2\,000$ .

### Nendetyra 3 [19 pike]

---

Ka ekzaktesisht dy guralece diku ne rreshtin 0: nje ne qelizen (0, x), tjetra ne qelizen (0, y); x dhe y jane te dallueshem dhe  $x + y$  eshte cift. Shkruani nje program qe len rrugematesin ne qelizen (0,  $(x + y) / 2$ ), p.sh., ekzaktesisht ne mesin ndermjet dy qelizave qe mbajne guralecet. Gjendja finale e grides nuk eshte e rendesishme.

*Limitet:* madhesia e programit  $\leq 100$ , gjatesia e ekzekutimit  $\leq 200\,000$ .

### Nendetyra 4 [deri ne 32 pike]

---

Ka te shumten 15 guralece ne gride, por nuk ka dy prej tyre ne te njejten qelize. Shkruani nje program qe l mbledh ato ne qoshen very-perendim ; ne menyre me precise nese ka x guralece ne gride ne fillim, ne fund duhet te jene ekzaktesisht x guralece ne qelizen (0, 0) dhe asnje guralec diku tjeter.

Piket per kete nendetyre varen nga gjatesia e ekzekutimit te programit te ngarkuar. Ne menyre me precise, nese L eshte maksimumi l gjatesise se ekzekutimit ne raste testimi te ndryshme, piket tuaja do te jene:

- 32 pike nese  $L \leq 200\,000$ ;
- $32 - 32 \log_{10} (L / 200\,000)$  pike nese  $200\,000 < L < 2\,000\,000$ ;
- 0 pike nese  $L \geq 2\,000\,000$ .

*Limitet:* madhesia e programit  $\leq 200$ .

### Nendetyra 5 [deri ne 28 pike]

---

Mund te jete cdo numer guralesesh ne cdo qelize te grides (ndermjet 0 dhe 15). Shkruani nje program qe gjen minimumin, p.sh qe perfundon me rrugematesin ne qelizen (i, j) ne menyre te tille qe cdo qelize tjeter te permbaje te pakten aq guralece sa (i, j). Pas ekzekutimit te programit, numri l guraleceve ne cdo qelize duhet te jete e njejte sa numri perpara ekzekutimit te programit.

Piket per kete nendetyre do te varen nga madhesia e programit P te programit te ngarkuar. Ne menyre me precise, piket tuaja do te jene:

- 28 pike nese  $P \leq 444$ ;
- $28 - 28 \log_{10} (P / 444)$  pike nese  $444 < P < 4\,440$ ;
- 0 pike nese  $P \geq 4\,440$ .

*Limitet:* gjatesia e ekzekutimit  $\leq 44\,400\,000$ .

### Detajet e implementimit

---

Duhet te ngarkoni ekzaktesisht nje skedar per nendetyre, te shkruar sipas rregullave te sintakses te specifikuar me siper. Cdo skedar l ngarkuar mund te kete madhesine maksimale prej 5 MiB. Per cdo

nendetyre, rrugematesi juaj do te testohet ne disa raste testimi dhe ju do te merrni informacion rreth burimeve te perdorura nga kodi juaj. Ne rastin kur kodi nuk eshte i sakte nga ana e sintakses dhe prandaj I pamundur per tu testuar ju do te merrni informacion rreth gabimit specific te sintakses.

Nuk eshte e nevojshme qe ngarkimi juaj te permbaje programet e rrugematesit per te gjitha nenedetyrat. Nese ngarkimi korrent nuk permban programin e rrugematesit per nendetyren X, ngarkimi juaj me I hershem per nendetyren X do te perfshihet automatikisht; nese nuk ka program te tille, nendetyra do te vleresohet me 0 per ate ngarkim.

Si zakonisht, piket e ngarkimit eshte shuma e pikeve te marra per cdo nendetyre dhe piket finale te detyres eshte shuma maksimale ndermjet ngarkimeve te testuara dhe ngarkimit te fundit.

### Simulatori

Per qellime testimi, ju jeni te pajisur me nje simulator rrugematesi, te cilin ju mund ta ushqeni me programet tuaja dhe griden e te dhenave. Programet e rrugematesit do te shkruhen ne te njejtin format te perdorur per ngarkimin(p.sh. ai i pershkruar me siper)

Pershkrimi I grides do te jipet duke perdorur formatin e meposhtem: cdo rresht I skedarit duhet te permbaje tre numra R, C dhe P, qe do te thote qe qeliza ne rreshtin R dhe kollonen C permban P guralece. Te gjitha qelizat te specifikuar ne pershkrimin e grides supozohen se nuk permbajne guralece. Per shembull konsideroni skedarin:

```
0 10 3
4 5 12
```

Grida e pershkruar nga ky skedar do te permbaje 15 guralece: 3 ne qelizen (0, 10) dhe 12 ne qelizen (4, 5).

Mund te therrisni simulatorin e testit duke thirrur programin `simulator.py` ne direktorin e detyrave, duke I kaluar emrin e skedarit si argument. Programi simulator do te pranoje opsionet e rreshtave te komandave si me meposhte:

- `-h` do te jape nje pershkrim te shkurter te opsioneve te gatshme;
- `-g GRID_FILE` ngarkon pershkrimin e grides nga skedari `GRID_FILE` (default: grida bosh);
- `-s GRID_SIDE` vendos madhesine e grides me `GRID_SIDE x GRID_SIDE` (default: 256, sic perdoret ne specifikimin e problemit); perdorimi I gridave te vogla mund te jete I dobishem per debugun e programit;
- `-m STEPS` limiton numrin e hapave te ekzekutimit ne simulim ne te shumten `STEPS`;
- `-c hyn` ne menyren e kompilimit; ne menyren e kompilimit, simulatori kthen ekzaktesisht te njejtin output, por ne vend qe te behet simulimi me Python, ai gjeneron dhe kompilon nje program te vogel ne C. Kjo shkakton nje ngarkese me te madhe kur fillon, por jep rezultate ne menyre sinjifikative me

te shpejta; keshilloheni ta perdorni ate kur programi juaj pritet te ekzekutohet per me shume se 10 000 000 hapa.

**Numri i ngarkimeve**

Numri maksimal i ngarkimeve te lejuara per kete detyre eshte 128.