

**Napomena:** Na Desktop-u se nalazi direktorijum `vi.jun1`. Preimenujte ga u oblik Vašeg indeksa `miGGBBB`. Na primer za indeks 283/2016 treba direktorijum `vi.jun1` preimenovati u `mi15283`. U ovom direktorijumu ostavite Vaše rešenje.

Na Desktop-u možete pronaći dokumentaciju za potrebne Python biblioteke.

Vreme za rad: 2 sata

1. Igrač u ruci ima tri karte. Ukoliko zameni jednu kartu dobiće kartu za 5 veću po vrednosti od zamenjene karte. Kako je najveća karta 14, deo vrednosti koja prelazi 14 računa se ciklično od 1. Na primer, za zamenjenu kartu 13, nova vrednost je 4 ( $14 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ ). Potrebno je korišćenjem A\* algoritma pronaći redosled zamena karata tako da igrač u ruci ima tri iste karte. Za heuristiku uzeti trenutni broj različitih karata u ruci. Zanimariti činjenicu da su 1 i 11 na kartama isti brojevi.

```
Ulaz:
Karte: [5,10,1]
Izlaz:
[[5, 10, 1], [5, 1, 1], [10, 1, 1], [1, 1, 1]]
```

Koristiti postavku `zadatak1.ipynb`

2. Vredni student Matematičkog fakulteta ostao je fasciniran genetskim algoritmima nakon predavanja na kursu Veštačke inteligencije. Zamislio je da u njegovom omiljenom jeziku Python implementira genetski algoritam koji će pokušati da od slike koja predstavlja proizvoljan šum koristeći samo pogodnu funkciju prilagođenosti generiše sliku koja predstavlja Python logo, ali ovaj, malo se namučio jer nije uspeo da isprati neke delove predavanja od umora. Pomozite mu!

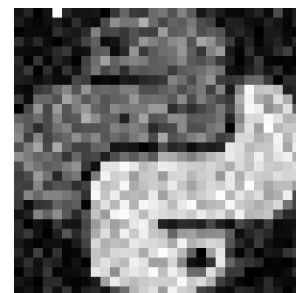
Slika 1 prikazuje originalnu sliku, slika 2 predstavlja sliku koja je skalirana tako da se smanji računaska zahtevnost i koja se koristi da se algoritam navodi, a slika 3 otprilike ono što bi vredni student želeo da dobije.



Slika 1: Originalna slika



Slika 2: Ciljna slika



Slika 3: Genetski algoritam

Koristiti postavku `zadatak2.ipynb`

3. U priloženoj datoteci za rad se nalazi implemetacija klase za zadavanje formula, implementacija KNF algoritma i deo implementacije DPLL algoritma.
  - (a) Dopuniti implementaciju DPLL algoritma implementacijom pravila *unit propagation*.
  - (b) Testirati rad DPLL nad formulom:  $(p \leftrightarrow q) \Rightarrow (\neg p \wedge r)$
  - (c) Napisati funkciju koja korišćenjem funkcije DPLL nalazi sve moguće valuacije koje zadovoljavaju formulu.
  - (d) Testirati rad napisane funkcije nad formulom:  $(p \vee q) \wedge (\neg p \vee \neg q)$

Koristiti postavku `zadatak3.ipynb`

4. Naš vredni student iz prethodnog zadatka nastavlja svoj vežbanje algoritama Veštačke inteligencije! Sada je rešio da popravi efikasnost izračunavanja algoritma A\* tako što će heurističku funkciju  $f$  aproksimirati na osnovu nekog atributa  $x$  tekućeg čvora linearnom regresijom jer je izračunavanje funkcije  $f$  u njegovom problemu izuzetno skupo. Potrebna mu je vaša pomoć tako što ćete implementirati sledeće korake:
  - (a) Vizualizujte podatke. Na  $x$  osi označite atribut  $x$ , a na  $y$  osi  $f(x)$ .
  - (b) Izvršiti podelu podataka na podatke za obučavanje i testiranje u razmeri 3:1.
  - (c) Koristeći modul `linear_model` napraviti model linearne regresije i obučiti ga na skupu podataka za obučavanje.
  - (d) Na standardni izlaz ispisati dobijene koeficijente modela (prikazati i slobodni član - eng. intercept).
  - (e) Ponovo nacratati podatke i na istoj slici nacrtati pravu dobijenu linearnom regresijom. Ose označiti isto kao u prethodnoj slici i dodati legendu koja prikazuje šta označava koja crtež na slici.
  - (f) Vizualizujte podatke. Na  $x$  osi označite atribut  $x$ , a na  $y$  osi  $f(x)$ .

Koristiti postavku `zadatak4.ipynb`