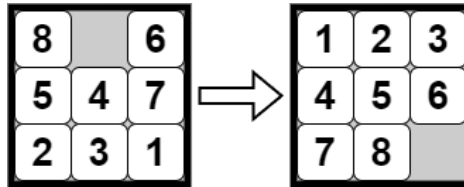




## Први пројектни задатак из предмета Интелигентни системи школске 2023/24. године

### Основне информације

**Puzzle** је графичка симулација написана на програмском језику Пајтон која приказује рад основних алгоритама претраживања на проблему игре *k-Puzzle*. Игру чини квадратна матрица плочица са ознакама од 1 до  $k$  и једном празнином са ознаком 0. Циљ игре је преуредити редослед плочица, померајући их хоризонтално или вертикално преко празнине, тако да она формирају матрицу са плочицама у растућем поретку.



Слика 1- Пример стартног и циљног стања за игру 8-Puzzle

У **Puzzle** симулацији плочице представљају делове учитане слике коју треба реконструисати. У ту сврху потребно је имплементирати неколико алгоритама и хеуристичких функција који решавају дати проблем.

### Начин коришћења апликације

Програм се покреће из терминала навођењем следеће команде:

```
.\main.py size image algoritm heuristic
```

где су:

- `.\main.py` – путања до главног Пајтон фајла са изворним кодом
- `size` – димензија матрице (подразумевано 3)
- `image` – назив слике која се користи (подразумевано `example.png`)
- `agent` – назив алгорита који се користи (подразумевано `ExampleAlgorithm`)
- `heuristic` – назив хеуристике која се користи (подразумевано `ExampleHeuristic`)

Пре покретања потребно је инсталирати пакет `pygame` у оквиру Пајтон интерпретера. Након покретања приказује се главни прозор апликације:



Слика 2 – Почетни екран



Приликом покретања програма, извршава се одабрани алгоритам, а потом приказује прозор апликације. У оквиру прозора приказана је матрица измешаних делова учитане слике над којом је могуће симулирати решење, а испод матрице исписан је тренутни корак симулације. Притиском на дугме *SPACE* могуће је покренути или привремено зауставити приказ решења. Притиском на дугме *ENTER* могуће је приказати коначно решење. Притиском на дугме *ESC* прекида се рад апликације и затвара се њен главни прозор.

## Алгоритми и хеуристике

Потребно је имплементирати следеће алгоритме:

- Претрага по ширини.
- Претрага прво-најбољи. Уколико постоје два или више чворова са једнаком вредношћу хеуристичке функције, такве чворове сортирати по растућој вредности идентификационе ознаке чвора\*.
- A\* алгоритам. Уколико постоје две или више парцијалних путања истих вредности функције процене, такве парцијалне путање сортирати по растућој вредности идентификационе ознаке последњег чвора на путањи.

Потребно је имплементирати следеће хеуристике:

- Hamming Број плочица које нису на одговарајућим циљним позицијама.
- Manhattan Сума Менхетн дистанци сваке плочице до одговарајуће циљне позиције.

\*идентификациона ознака чвора – торка целих бројева ознака плочица добијена линеаризацијом матрице по врстама редом одозго на доле (нпр. за стања са слике 1 (8, 0, 6, 5, 4, 7, 2, 3, 1) је идентификациона ознака за старно, а (1, 2, 3, 4, 5, 6, 7, 8, 0) за циљно стање).

## Кориснички захтеви

Студенти треба да имплементирају стратегије претраге проширивањем класе `Algorithm` (чија дефиниција треба да се нађе у датотеци `algorithms.py`) и редефинисањем њене функције `get_steps`, чији параметри `initial_state` и `goal_state` представљају идентификационе ознаке стартног и циљног стања игре, а повратна вредност је коначна листа акција које доводе до циљног стања. Она је представљена листом индекса, где сваки индекс представља позицију у идентификационој ознаци (индексирану од 0) за плочицу која се помера на празну плочицу (нпр. за стартно стање са слике 1, померање плочице 6 је задато акцијом 2, јер се она налази на позицији 2 у идентификационој ознаци стања). Студентима је у оквиру кода дат пример класе (`ExampleAlgorithm`) која проширује основну класу `Algorithm` и на једноставан начин имплементира функцију `get_steps`.

У наставку је дат пример покретања програма са датом пример класом:

```
.\main.py 3 example.png ExampleAlgorithm
```

Имати у виду да се на одбрани домаћег задатка могу појавити нови алгоритми и хеуристике.

Имплементација се ради у програмском језику Пајтон уз дозвољено коришћење свих стандардних библиотека и структура података.



Универзитет у Београду - Електротехнички факултет  
Катедра за рачунарску технику и информатику

### Напомене

Електронску верзију решења овог домаћег задатка послати најкасније до 16. новембра 2023. године до 15 часова, као ZIP архиву на следећем [линку](#). Одбрана домаћег задатка планирана је 17. новембра 2023. године.

Домаћи задатак из предмета *Интелигентни системи* се ради самостално и није обавезан за полагање испита (на испиту се може заменити са теоријским питањима из целокупног градива). Овај домаћи задатак се може бранити само у за то предвиђеном термину. Домаћи задатак вреди максимално 10 поена.

На усменој одбрани кандидат мора самостално да покрене своје решење. Кандидат мора да поседује потребан ниво знања о задатку, мора да буде свестан недостатака приложеног решења и могућности да те недостатке реши. Кандидат мора тачно да одговори и на одређен број питања која се баве тематиком домаћег задатка.

Евентуална питања послати асистентима на мејл, али као једну поруку, а не две одвојене (другог асистента обавезно ставити у копију - СС поруке).

aki@etf.rs

jocke@etf.rs