



## Домашна работа 3

*курс Структури от данни и програмиране  
за специалност Информатика  
зимен семестър 2018/19 г.*

Нека е даден текстов файл, който описва картата, по която се развива действието в компютърна игра. Картата се състои от една или повече зони. Във всяка от зоните може да има преход към една или повече от останалите. Например:

```
[zones]
village -> dock [iron key];
village -> basement;
basement -> village;
dock -> village;
dock -> jungle;
jungle -> temple_entrance;
temple_entrance -> temple_1a [skull key];
temple_entrance -> temple_2 [serpent key];
temple_1a -> temple_1b;
desert -> oasis;
```

```
[keys]
iron key -> basement;
iron key -> desert;
skull key -> temple_2;
serpent key -> temple_1b;
```

Секцията [zones] описва всички дъги в графа. Както се вижда от примера, той е насочен. Веднъж напуснали дадена зона, не винаги можем да се върнем в нея или поне не винаги директно.

В някои от зоните са скрити ключове. Това е описано в секцията [keys], в която за всеки ключ е указано къде се намира. Възможно е копия на даден ключ да се намират в няколко различни зони.

Ключовете имат етикети. Когато една дъга в секцията [zones] е маркирана с етикет К, това означава, че по нея може да се премине само и единствено ако играчът притежава ключ с етикет К. В противен случай по тази дъга не може да се преминава. Например, за да преминем от village към dock, трябва да имаме ключ iron key.

Вашата програма ще получи като вход (1) път до текстов файл в описания формат, който съдържа карта и (2) името на зоната, от която започва играта. Можем да считаме, че файлът ще бъде синтактично коректен. Програмата ви трябва да провери дали в картата няма недостижими зони. Това са:

- Зони, до които не съществува път започващ от началната зона. В примера, такива са desert и oasis.
- Зони, които са достижими с ключ, който обаче няма как да се вземе тръгвайки от началната зона. В примера, такива зони са temple\_1a, temple\_1b и temple\_2. За да бъдат достигнати temple\_1a и temple\_1b е нужен ключ skull key. Той е скрит в зоната temple\_2, достижима с ключ serpent key. Той обаче е скрит в temple\_1b. Така тези зони се "заклучват" взаимно.

Всички останали зони в картата от примера са достижими. След като програмата ви анализира картата за коректност, тя трябва да генерира текстов файл в т.нар. [DOT формат](#), в който да запише резултатите от проверката. Такъв файл може да се визуализира с помощта на приложението [Graphviz](#). Когато решавате задачата, можете да използвате Graphviz, за да визуализирате файловете, които генерира програмата ви и така по-лесно да проверите дали тя дава коректен резултат. Може да използвате и някой онлайн редактор/визуализатор, като например:

<http://www.webgraphviz.com/>  
<https://dreampuf.github.io/GraphvizOnline/>  
<http://viz-js.com/>

Изходният файл трябва да съдържа представяне на картата като граф. В него:

- Преходите, които изискват играчът да притежава даден ключ трябва да са надписани с името на ключа.
- Ако дадена зона съдържа един или повече ключове, те трябва да се добавят към името ѝ.
- Недостижимите зони трябва да бъдат маркирани в червено.

По-долу е показано как може да изглежда изходният файл за картата от примера, а също и как ще изглежда графът генериран от Graphviz.

```
digraph {
  village -> dock [label="iron key"];
  village -> basement;

  dock -> village;
  dock -> jungle;

  basement[label="basement\liron key"];
  basement -> village;

  jungle -> temple_entrance;

  temple_1a[color=red,style=filled,fillcolor="#ffefef"];
  temple_1b[label="temple_2\lserpent key",color=red,style=filled,
fillcolor="#ffefef"];
  temple_2[label="temple_2\lskull key",color=red,style=filled,
fillcolor="#ffefef"];

  temple_entrance -> temple_1a [label="skull key"];
  temple_1a -> temple_1b;
  temple_entrance -> temple_2 [label="serpent key"];

  desert[label="desert\liron key", color=red, style=filled, fillcolor="#ffefef"];
  oasis[color=red,style=filled, fillcolor="#ffefef"];
  desert -> oasis;
}
```

