

Hledání nejkratší cesty v bludišti

24. února 2023

1 Textový popis

Tento projekt se zabývá řešením (hledáním nejkratší cesty) a také základním generováním bludišť. Základním vstupem bude bludiště $n \times n$, přičemž vstup do bludiště bude vždy levý horní roh a výstup bude vždy pravý dolní roh. Z jedné buňky do druhé se lze dostat pouze přes společnou hranu (nikoliv přes roh). Cílem projektu je implementovat algoritmy pro načítání, hledání nejkratší cesty a generování bludiště.

Na začátku bude implementována funkce pro načítání bludiště z CSV souboru. Tato funkce bude umět načítat bludiště o libovolném rozměru $n \times n$ a uložit ho do paměti v podobě NumPy matice s True/False hodnotami (True = buňka je neprůchozí). Poté bude implementován algoritmus pro hledání nejkratší cesty. Poslední částí bude vytvoření generátoru bludiště za použití algoritmu pro hledání nejkratší cesty.

Výstup bude formou obrázku (černá = neprostupná část, bílá = průchozí, červená = nejkratší cesta).

2 Funcionality

- Implementovat načítání bludiště z CSV souboru
- Implementovat algoritmus pro hledání nejkratší cesty (mezi levým horním rohem a pravým dolním rohem) za použití knihovny NumPy, který bude pracovat v následujících dvou krocích:
 - Sestavení incidenční matice
 - Hledání nejkratší cesty pomocí Dijkstrova algoritmu
- Zapsat bludiště a nalezenou cestu do černobílého obrázku, kde cesta bude vyznačena červeně
- Vytvořit funkci pro generování bludiště tak, aby mělo řešení (tj. aby existovala cesta mezi levým horním a pravým dolním rohem)
 - funkce začne s nějakou šablonou (předdefinované v kódu) a poté bude zaplňovat bludiště v náhodných místech a kontrolovat, zda je stále průchozí
 - šablon bude více (např. empty = volné bludiště, slalom = bariéry aby cesta musela minimálně mít tvar S, ...) - budou s obrázky ukázané v Readme

A Další povinné části projektu společné pro všechny projekty

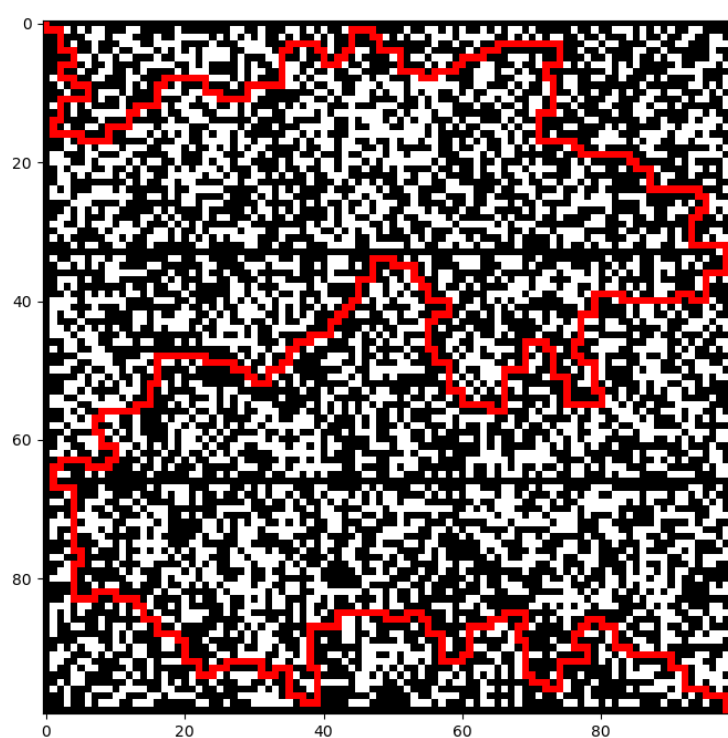
- struktura projektu:
 - složka s řešením = knihovnou
 - složka s testovacími daty
 - Jupyter notebook obsahující demonstraci funkcionalit/examples
 - Readme file obsahující popis projektu
- Dokumentace
 - type hinting - minimálně pro všechny funkce, které jsou v tzv. uživatelském rozhraní (uživatel je používá, pokud používá knihovnu)
 - rozumné docstringy popisující chování fanných funkcí a tříd
 - názvy funkcí a tříd v souladu se zavedenou domluvou (snake_case, ...)
- Provedení
 - rozumné využití OOP (zde například, objektem by mělo být samotné bludiště a funkce s ním pracující, jeho metody)
 - rozumná délka funkcí - každá funkce má právě jednu zodpovědnost, nekupit funkcionality
 - nekopírovat části kódu
 - nepoužívat tzv. hardcoded konstanty

B Obecná náročnost projektu

Projekt by měl obsahovat typově tyto části:

- načítání zadání ze souboru
- vykreslování řešení a jeho zapisování do souboru
- řešení problému, který není triviální
 - vyžaduje nějaký algoritmus (nemusí být nutně matematický nebo složitý)
 - ideálně více kroků v různých funkcích, případně více různých funkcionalit

C Vzorová ukázka výstupu vzorového zadání projektu



Obrázek 1: Ukázka vyřešeného bludiště