Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

SEMESTRÁLNÍ PRÁCE z UZI

Přelévání kapaliny se zdrojem a kanálem

24. 12. 2024 Miloš Barborka

Obsah

[1. Zadání 2](#_Toc185971942)

[2. Analýza úlohy 2](#_Toc185971943)

[3. Popis algoritmu řešení 2](#_Toc185971944)

[4. Popis programu (programová dokumentace) 3](#_Toc185971945)

[1. Data 3](#_Toc185971946)

[2. Inference 3](#_Toc185971947)

[3. Komunikace 3](#_Toc185971948)

[4. Vysvetlovani 4](#_Toc185971949)

[5. Znalosti 4](#_Toc185971950)

[6. Main.py 4](#_Toc185971951)

[5. Popis obsluhy programu (uživatelská dokumentace) 4](#_Toc185971952)

[6. Rozbor výsledků, zhodnocení 5](#_Toc185971953)

[7. Závěr 5](#_Toc185971954)

# Zadání

Je dáno N (N >= 2 ˄ N <= 4) nádob, jejich objem, počáteční naplnění a koncové

naplnění. Vytvořte program, který najde nejkratší posloupnost přelití vedoucí

z počátečního stavu do koncového stavu a pro přelévání je k dispozici nevyčerpatelný

zdroj kapaliny a nepřeplnitelný kanál.

# Analýza úlohy

• Alternativy řešení:

Použití BFS (prohledávání do šířky) pro systematické prohledávání stavového prostoru.

Heuristický algoritmus A\* pro zkrácení výpočtového času při větších prostorech.

• Výběr:

BFS byl zvolen pro svou jednoduchost a záruku nalezení nejkratší cesty.

# Popis algoritmu řešení

1. Inicializace: Definice počátečního stavu, fronty a množiny navštívených stavů.
2. Iterace:
   * Odeber stav z fronty.
   * Generuj možné přelití.
   * Ulož nové stavy, pokud ještě nebyly navštíveny.
3. Ukončení: Pokud je nalezen cílový stav, algoritmus vrátí sekvenci kroků.

**Schéma:**  
Počáteční stav → Generování přelití → Kontrola cílového stavu → Aktualizace fronty → Opakování.

# Popis programu (programová dokumentace)

 Technologie:

* Jazyk: Python (jednoduchost, dostupnost knihoven).
* Knihovny: collections (deque) pro efektivní fronty.

 Datové struktury:

* Seznamy: Pro reprezentaci aktuálního stavu nádob.
* Slovníky: Pro ukládání již navštívených stavů.

 Popis hlavních modulů:

1. Data
2. Inference
3. Komunikace
4. Vysvetlovani
5. Znalosti
6. Main.py

## Data

Zde se nachází ještě složka s akcemi (naplneni, preliti, vyliti a tah). Tah je zde pouze jako abstraktní třída pro druhy akcí, aby se s nimi lépe manipulovalo.

Dále data obsahují třídy nádoba, stav\_nadob. Nadoba zprostředkovává jednotlivé nádoby a výpis co se v nádobě nachází. Stav\_nadob slouží ke zjištění určitého stavu nádoby.

## Inference

Obsahuje inferencni\_modul, tento modul se stará o chod programu, získávání posloupnosti stavů, přes které se procházelo, také generuje stavy a neposlední řadě je přidává do fronty.

## Komunikace

Obsahuje komunikacni\_modul, tento modul má za úkol komunikovat s uživatelem. Je zde vypisování běh programu, jsou zde kladeny otázky ohledně naplnění nádob, či konkrétního počtu nádob, které půjdou do programu. Také se zde kontroluje přijímaný vstup od uživatele.

## Vysvetlovani

Zde se vyskytuje vysvetlovaci\_modul, tento modul se stará aby uživatel věděl jak program došel do jednotlivého koncového stavu, takže mu vypisuje posloupnost a také vysvětluje proč se posloupnost stavů nevypsala (způsobeno tím, že se z nějakého stavu do další nedá dostat.)

## Znalosti

Obsahuje pouze akce\_nadoby a zpracovane\_stavy. Akce nadob určuje, co se s určitou nádobou udělá, zda se přelije její obsah do jiné, naplní se do plna nebo vylije její obsah do kanálu. Zpracovane stavy pouze kontroluje, zda fronta obsahuje daný stav nebo stav přidá.

## Main.py

Třída, která spouští celý program.

# Popis obsluhy programu (uživatelská dokumentace)

Program se dá spouštět přes příkazovou řádku, když si uživatel stáhne všechny složky. Nejsou třeba dodávat žádné jiné knihovny.

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automatickyPříklad zadávání hodnot jednotlivých nádob viz. obrázek č. 1

Obrázek Zadávání vstupů

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automatickyPříklad výstupu pro zadané hodnoty z obrázku č. 1

Obrázek Výstupní posloupnost

Výstup programu, když se nepodaří nalézt cestu:

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automatickyProgram vysvětlení používá na dotázání jen pří nenalezení cesty, jelikož když se cesta nalezne tak to je vlastně to celé vysvětlení.

Obrázek Nenalezená cesta

# Rozbor výsledků, zhodnocení

Myslím si, že jsem nalezl optimální řešení pro tento problém a také by daný program měl

být optimalizovaný čímž bych chtěl eliminovat časovou složitost. Zároveň při použití

prohledávání BFS (prohledávání do šířky) tak naleznu nejkratší cestu k cíli.

Jako nedostatky bych možná dal možná volbu různých jednotek, a i následné přepočty.

Můj program je totiž řešen pouze obecně, ale myslím si, že by to nebylo těžké dodat, je to

spíše něco navíc.

# Závěr

Možná zlepšení jsem již zahrnul ve zhodnocení, ale například by šlo ještě třeba nádoby vizualizovat ve 2D, i jejich přelévání. Vypadalo by to velice pěkně a bylo by to více názorné. Obecně vzato si myslím, že je úloha splněna.