

# Algorytmy dla Problemów Trudnych Obliczeniowo

## Projekt 2024: Wyjazd Dziekana

### Problem

Prof. Lasowski, dziekan nowopowstałego wydziału na pewnej znanej uczelni chce dojechać na swój dyżur dydaktyczny. Niestety okazało się, że jego samochód został zablokowany przez inne na parking. Samochody okazały się nie mieć zaciągniętych hamulców ręcznych i Dziekan postanowił przepchnąć je tak, żeby mógł wyjechać. Niestety przepchnięcie każdego samochodu wymaga pewnej energii, a Dziekan chce mieć jak najwięcej sił na spotkania ze studentami. Stąd najpierw postanowił stworzyć program, który wskaże w jakiej kolejności i które samochody ma przesunąć tak, żeby wykonać najwyżej zadaną liczbę ruchów (przez “stworzyć” Dziekan rozumie dać możliwość wykazania się rzeczonym studentom). Zadanie polega na rozwiązaniu problemu Dziekana.

Wejście stanowi opis planszy o rozmiarach  $W \times H$  pól ( $W$  to szerokość,  $H$  to wysokość planszy). Na każdym polu planszy znajduje się jeden znak oznaczający albo wolne pole, albo nieprzesuwalną blokadę, albo samochód. Każdy samochód zajmuje od 2 do 4 pól i albo jest zorientowany poziomo albo pionowo. Każdy samochód można przesunąć wyłącznie zgodnie z jego orientacją. Jeden ruch polega na przesunięciu jednego samochodu o dowolną liczbę pól w zadanym kierunku (góra/dół dla orientacji pionowej, prawo/lewo dla orientacji poziomej) ale tak, że w żadnym momencie samochód nie może zajmować pola, na którym jest już inny samochód lub blokada. Samochód Dziekana może wyjechać z parkingu w momencie gdy można go przesunąć poza obszar parkingu. Żaden inny samochód nie może opuścić terenu parkingu, ani nawet znaleźć się częściowo poza nim. Celem jest doprowadzenie do wyjazdu Dziekana wykonując najwyżej  $N$  ruchów (ostatni ruch, w którym Dziekan opuszcza parking także należy uwzględnić w tych  $N$  ruchach; co najmniej jeden fragment samochodu Dziekana musi się znaleźć poza planszą).

### Wejście

Wejście składa się z jednego wiersza zawierającego po kolei liczby  $W$ ,  $H$ , oraz  $N$ , po którym następuje ciąg  $H$  wierszy, gdzie każdy zawiera  $W$  znaków opisujących kolejne pola planszy, o następującym znaczeniu:

1. "." – puste pole,
2. "#" – pole z blokadą,
3. "a" – początek samochodu zorientowanego poziomo, kolejne pole na prawo zawiera symbol "b", dalsze pola mogą zawierać "c" i dalej "d" (auta o długości 2 kończą się na "b", auta o długości 3 kończą się na "c", a te o długości 4 kończą się na "d").

4. "x" – początek samochodu zorientowanego pionowo, kolejne pole w dół od tego zawiera symbol "y", dalsze pola mogą zawierać "z" i dalej "w" (auta o długości 2 kończą się na "y", auta o długości 3 kończą się na "z", a te o długości 4 kończą się na "w").
5. "o" – pole zajmowane przez samochód Dziekana (samochód Dziekana składa się z od 2 do 4 symboli "o" ułożonych po kolei pionowo lub poziomo, zgodnie z jego orientacją; na planszy jest tylko jeden samochód Dziekana).

**Przykład.** Poniżej znajduje się opis przykładowego zadania:

```
6 5 4
#####
#..ab#
#..xx#
#ooyy.
#####
```

Na tej planszy znajdują się cztery samochody. Samochód Dziekana (orientacja pozioma), jeden dodatkowy samochód o orientacji poziomej, oraz dwa samochody o orientacji pionowej. Poniżej przedstawiamy graficzną reprezentację planszy (samochód Dziekana jest czerwony):

	0	1	2	3	4	5
0						
1						
2						
3						
4						

## Wyjście

Pierwszy wiersz wejścia powinien zawierać liczbę  $M$  ruchów, które zawiera rozwiązanie ( $M \leq N$ ; mogą być wejścia, dla których istnieją krótsze rozwiązania niż  $N$ ). Następnie wyjście powinno zawierać  $M$  wierszy postaci  $X \ Y \ D \ L$ , opisujących kolejne ruchy:

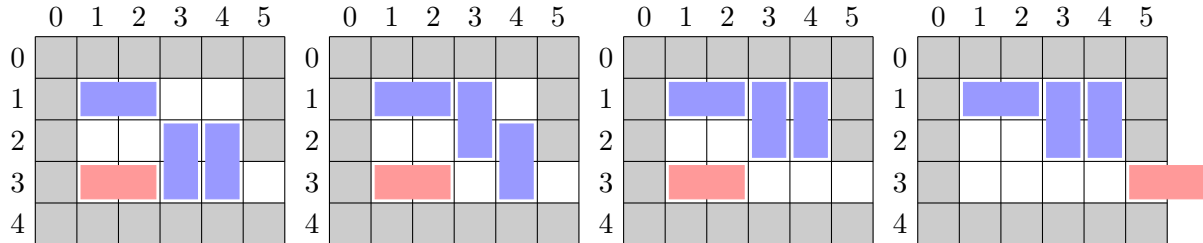
1.  $X \ Y$  to współrzędne pola, na którym znajduje się przesuwany w danym ruchu samochód (można wskazać dowolne pole zajmowane przez ten samochód; lewy górny narożnik planszy ma współrzędne 0,0).
2.  $D$  to kierunek ruchu (U to ruch w górę, D to ruch w dół, L to ruch w lewo, R to ruch w prawo; ruch musi być zgodny z orientacją samochodu).
3.  $L$  to liczba pól, o które należy przesunąć samochód (przesunięcie nie może prowadzić do kolizji z innym samochodem lub blokadą).

Ostatni ruch powinien doprowadzić do sytuacji, że samochód Dziekana znajdzie się co najmniej jednym swoim elementem poza polem parkingu.

**Przykład.** Przykładowym rozwiązaniem powyższej planszy jest następujące wyjście:

```
4
3 1 L 2
3 3 U 1
4 2 U 1
2 3 R 4
```

Poniżej przedstawiamy plansze po kolejnych ruchach:



## Zadanie

Proszę zaimplementować program w języku C/C++, który wczytuje ze standardowego wejścia opis parkingu oraz wypisuje na standardowe wyjście opis rozwiązania (można założyć, że rozwiązanie zawsze istnieje). Program powinien spełniać następujące warunki:

1. Program jest jednowątkowy i jednoprosesowy (nie wykorzystuje w żaden sposób mechanizmów wielowątkowości).
2. Program nie odwołuje się do żadnych operacji wejścia/wyjścia (w tym dostępu do sieci i plików) innych niż czytanie ze standardowego wejścia i zapisywanie na standardowe wyjście.