

# Algorytmy i struktury danych

## Projekt zaliczeniowy

mgr inż. Bartłomiej Kizielewicz, mgr inż. Andrii Shekhovtsov

14 stycznia 2023

## 1 Zasady oceniania

Ocena za projekt jest obliczana zgodnie z punktacją pokazaną w Tabeli 1. W przypadku częściowo zrobionego zadania mogą być naliczone punkty częściowe.

Tabela 1: Punkty za projekt.

Wykonana dobrze część projektu	Ocena
Zaimplementowany 1 scenariusz i 1 algorytm, przedstawienie wyniku symulacji dla tego przypadku.	3.0
Zaimplementowane 3 scenariusze i 1 algorytm, przedstawione wyniki symulacji oraz średnia odległość z 3 scenariuszy.	3.5
Zaimplementowane 3 scenariusze i 2 algorytmy, przedstawione wyniki symulacji oraz wartości średnia dla tych przypadków.	4.0
Jak wyżej (4.0) + wnioski.	4.5
Zaimplementowane 3 scenariusze i 3 algorytmy, przedstawione wyniki symulacji oraz wartości średnie dla tych przypadków, podane wnioski, algorytm 3 jest opisany w sposób zrozumiały.	5.0

W wyniku wykonania projektu mają powstać dwa pliki:

- Plik w formacie **PDF** z raportem, które ma zawierać schematy blokowe, tabele z wynikami, wnioski i inne wyniki wykonania projektu które nie są kodem.
- Plik w formacie **.py** który ma zawierać kod w języku Python który powstał w ramach tego projektu, czyli kod symulacji, własnego algorytmu poruszania się windy, itd. Poszczególne sekcje tego pliku muszą być sensownie skomentowane.

Pliki te należy przesłać za pośrednictwem systemu Moodle we wskazanym tam terminie.

*Wysłanie projektu w formie niezgodnej z powyższą może skutkować wystawieniem oceny niedostatecznej.*

Termin oddania raportu jest ustawiony w systemie Moodle. W przypadku nie oddania zadania w terminie, uzyskana ocena będzie zmniejszana o 0,5 za każdy zaczęty tydzień opóźnienia.

## 2 Ogólny opis projektu

Należy zbadać 3 algorytmy poruszania się windy w 3 różnych sytuacjach. Za pomocą symulacji (szczegółowy opis poniżej) należy zbadać który z tych algorytmów poruszania się windy będzie najlepszy pod kątem pokonywanej przez windę odległości.

Kilka definicji:

- **Scenariusz** - sposób losowania piętr, może być przykładowo zaimplementowany jako funkcja które zwraca dwie wartości: piętro początkowe i piętro końcowe (docelowe). Scenariusze są oznaczane zwykłymi liczbami 1, 2, 3.
- **Algorytm** - sposób w jaki zachowuje się winda, przykładowo może być to zaimplementowane jako funkcja która zapisuje do listy („historii” poruszania się windy) numery piętr w określony sposób. Algorytmy oznaczane są liczbami rzymskimi (I, II, III).

### 3 Szczegółowy opis

Chcemy zbadać jaki z 3 algorytmów będzie najlepszy pod kątem pokonywanej w budynku odległości (im mniej pokonuje tym lepiej, bo dłużej się zużywa).

Założenia ogólne:

- Blok mieszkalny ma 10 pięter, czyli 11 kondygnacji (liczymy od 0, wtedy 0 będzie parterem + 10 kolejnych kondygnacji).
- Wysokość jednej kondygnacji to 2.8m, czyli jeżeli winda wjechała z parteru (0) na 1 piętro, to pokonała ona 2.8m.
- Dla każdego z 9 przypadków (3 algorytmy razy 3 scenariusze) wykonujemy 1000 powtórzeń, czyli 1000 losowań piętro początkowe - piętro końcowe.

Należy przebadać 3 algorytmy poruszania się windy w **3 scenariuszach** (oznaczona zwykłymi liczbami):

1. Całkowicie losowe podanie piętra początkowego i piętra końcowego. Przykładowo, można to zaimplementować w ten sposób (należy odpowiednio wcześniej zaimportować bibliotekę `random`):

```
1 def script1():
2     start = random.randint(0, 10)
3     finish = random.randint(0, 10)
4
5     # Zapewniamy że start i finish są różne
6     while start == finish:
7         finish = random.randint(0, 10)
8
9     return start, finish
10
11 # Możemy użyć tej funkcji w dalszym kodzie w taki sposób:
12 start, finish = script1()
```

2. Symulujemy przypadek kiedy ludzie jadą do pracy: 70% ludzi zjeżdża z pięter powyżej 4 na parter. W pozostałych 30% wypadków stosujemy scenariusz 1. Przykładowa implementacja:

```
1 def script2():
2     if random.random() < 0.7:
3         start = random.randint(4, 10)
4         finish = 0
5
6         return start, finish
7     return script1()
8
9 # Możemy użyć tej funkcji w dalszym kodzie w taki sposób:
10 start, finish = script2()
```

3. Symulujemy przypadek kiedy ludzie wracają z pracy/zakupów: 70% ludzi jedzie z parteru (0) na dowolne inne piętro. W pozostałych 30% wypadków stosujemy scenariusz 1. Przykładowa implementacja:

```
1 def script3():
2     if random.random() < 0.7:
3         start = 0
4         finish = random.randint(1, 10)
5
6         return start, finish
7     return script1()
8
9 # Możemy użyć tej funkcji w dalszym kodzie w taki sposób:
10 start, finish = script3()
```

W kolejnych sekcjach są przedstawione **3 algorytmy** (oznaczone liczbami rzymskimi) poruszania się windy które należy przebadac. Każdy algorytm należy przebadac w każdym z 3 scenariuszy.

(I) Winda zostaje na tym piętrze na którym skończyła swój ruch.

Rozpatrzmy przykład dla 3 przejazdów windy, w tabeli poniżej podane piętra początkowe oraz końcowe:

Start	Finish
0	7
2	5
6	1

Przy takim losowaniu w tym algorytmie winda będzie zatrzymywać się na następujących piętrach:

$$0 \rightarrow 7 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 1$$

Przebytą odległość liczymy jako różnice bezwzględną pomiędzy kolejnymi piętrami razy wysokość kondygnacji, czyli:

$$(|0 - 7| + |7 - 2| + |2 - 5| + |5 - 6| + |6 - 1|) \times 2.8 = \\ = (7 + 5 + 3 + 1 + 5) \times 2.8 = 58.8$$

Czyli korzystając z tego algorytmu winda pokonała odległość 58.8 m.

(II) Winda po przejeździe zawsze wraca na parter (piętro 0), a dopiero potem jedzie na kolejne piętro.

Rozpatrzmy ten sam przykład co dla poprzedniego algorytmu:

Przy takim losowaniu w tym algorytmie winda będzie zatrzymywać się na następujących piętrach:

$$0 \rightarrow 7 \rightarrow 0 \rightarrow 2 \rightarrow 5 \rightarrow 0 \rightarrow 6 \rightarrow 1 \rightarrow 0$$

Czyli pokonana odległość będzie wynosić:

$$(7 + 7 + 2 + 3 + 5 + 6 + 5 + 1) \times 2.8 = 100.8$$

Czyli korzystając z tego algorytmu winda pokona odległość 100.8m.

(III) Algorytm autorski - należy opracować własny algorytm poruszania się windy oraz opisać go za pomocą schematu blokowego oraz słownie.

W wyniku takiego badania ma powstać tabela z wynikami, przykład takiej tabeli poniżej. Poszczególne pola tej tabelki muszą zawierać odległość którą przebyła winda po symulacji przejazdu windą 1000 ludzi przy użyciu wybranego scenariusza i wybranego algorytmu. Proszę skonwertować te wartości w kilometry i zaokrąglić do dwóch wartości po przecinku. Wartości podane w tabeli są przykładowe i nie należy na nie się wzorować.

Scenariusz	Algorytm I	Algorytm II	Algorytm III
Scenariusz 1	10.21 km	8.78 km	7.38 km
Scenariusz 2	...	...	...
Scenariusz 3	...	...	...
Wartość średnia	...	...	...

Na podstawie powyższej tabeli z wynikami należy wyciągnąć odpowiednie wnioski. Należy wskazać który z algorytmów pasuje do którego scenariusza oraz który z nich jest lepszy jeżeli uwzględnić tylko średnią odległość.