

Projekt inżynierski

Jakub Maternia

6 stycznia 2025

Spis treści

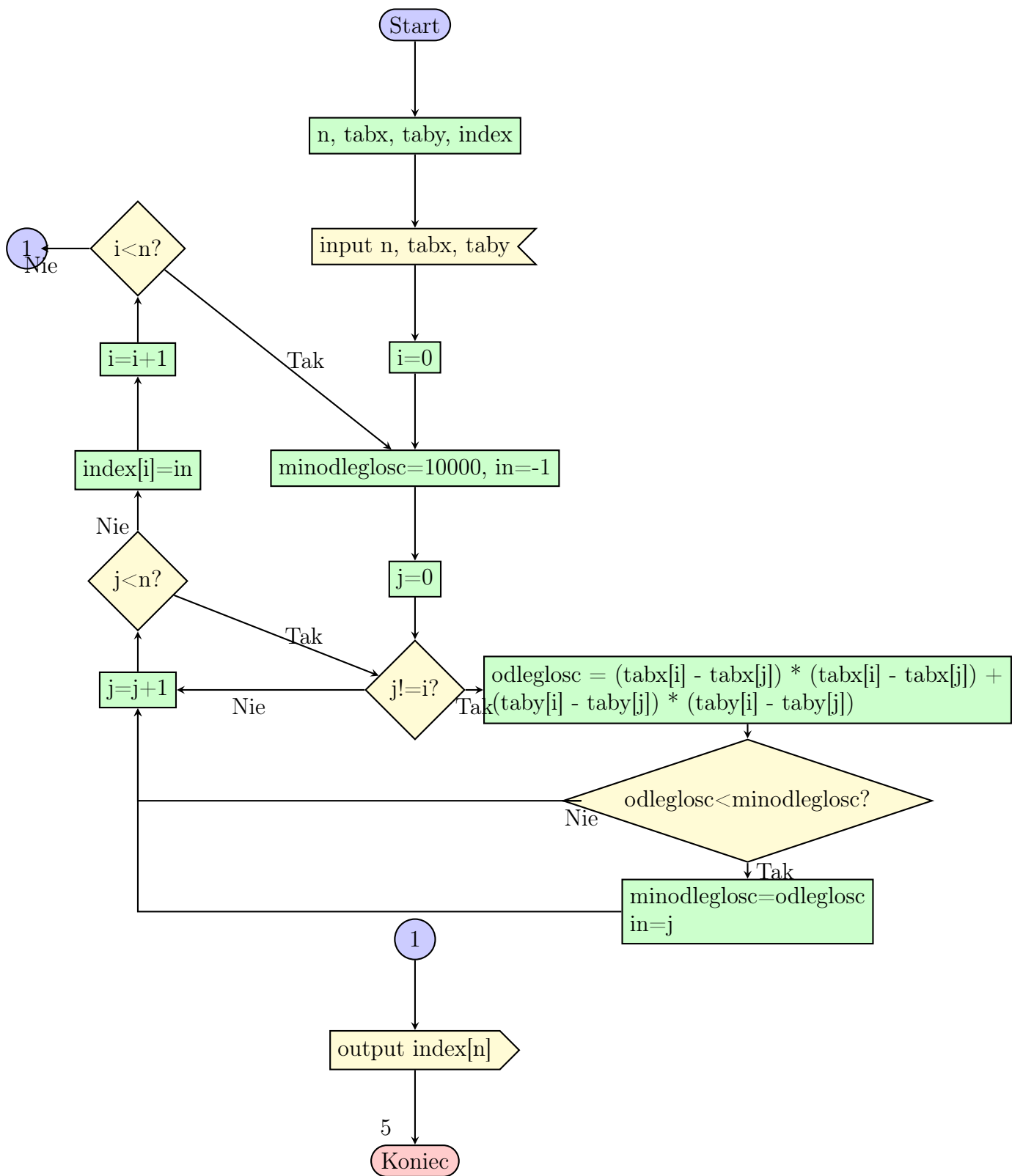
1	Treść zadania	3
2	Podjęcie brute force	3
2.1	Analiza problemu i potencjalne rozwiązanie	3
2.2	Schemat blokowy	4
2.3	Pseudokod	6
2.4	Przykładowe rozwiązanie	7
2.5	Złożoność obliczeniowa	7
2.6	Implementacja	7
3	Druga metoda z wykorzystaniem przestrzeni	7
3.1	Metoda działania	7
3.2	Schemat blokowy	7
3.3	Pseudokod	7
3.4	Przykładowe rozwiązanie	7
3.5	Złożoność obliczeniowa	7
3.6	Implementacja	7
4	Implementacja obu algorytmów	7
4.1	Testy wydajności	7
4.1.1	Metoda Brute force	7
4.1.2	Druga metoda	7

1 Treść zadania

2 Podejście brute force

2.1 Analiza problemu i potencjalne rozwiązanie

2.2 Schemat blokowy



2.3 Pseudokod

1. input: n, tabx, taby
2. output: index
3. index := -1
4. dla i od 0 do n-1 wykonaj
5. minodleglosc := 10000
6. in := -1
7. dla j od 0 do n-1 wykonaj
8. jeśli i != j wtedy
9. odleglosc := (tabx[i] - tabx[j]) * (tabx[i] - tabx[j]) + (taby[i] - taby[j])
* (taby[i] - taby[j])
10. jeśli odleglosc < minodleglosc wtedy
11. minodleglosc := odleglosc
12. in := j
13. koniec jeśli
14. koniec jeśli
15. koniec dla
16. index[i] := in
17. koniec dla
18. zwroc index

2.4 Przykładowe rozwiązanie

2.5 Złożoność obliczeniowa

2.6 Implementacja

3 Druga metoda z wykorzystaniem przestrzeni

3.1 Metoda działania

3.2 Schemat blokowy

3.3 Pseudokod

3.4 Przykładowe rozwiązanie

3.5 Złożoność obliczeniowa

3.6 Implementacja

4 Implementacja obu algorytmów

4.1 Testy wydajności

4.1.1 Metoda Brute force

4.1.2 Druga metoda