Pierwsza Ławka · zestaw działowy · liceum i technikum

Dział 11

Podejście zachłanne i programowanie dynamiczne; podciągi

Wydawanie reszty i najkrótsza droga (algorytm Dijkstry) metodą zachłanną, programowanie dynamiczne (najmniejsza liczba monet, najdłuższy wspólny podciąg), najdłuższy spójny podciąg niemalejący i spójny podciąg o największej sumie. Osiem kart: teoria z przykładami, ćwiczenia, utrwalenie w formule CKE, zadanie przy komputerze z plikami danych (ZIP), kartkówka i sprawdzian w grupach A i B – z kluczem i zasadami oceniania.

Informatyka | klasa 4 LO i technikum | Dział 11 | zakres rozszerzony (ZR)

	Karta	Rodzaj	Poziom	Zadania
1	**Podejście zachłanne i programowanie dynamiczne; podciągi – najważniejsze wiadomości** Teoria, przykłady krok po kroku, rozgrzewka	Najważniejsze wiadomości	ZR	4 zad. · 7 pkt
2	**Podejście zachłanne i programowanie dynamiczne; podciągi – ćwiczenia podstawowe** Ćwiczę umiejętności z działu	Ćwiczenia	ZR	7 zad. · 20 pkt
3	**Podejście zachłanne i programowanie dynamiczne; podciągi – utrwalenie** Zadania w formule CKE: polecenia i punktacja jak w arkuszach maturalnych	Zadania typu maturalnego	ZR	7 zad. · 13 pkt
4	**Podejście zachłanne i programowanie dynamiczne; podciągi – w praktyce, przy komputerze** Pliki danych w archiwum ZIP · wyniki w plikach wynikiN.txt	Ćwiczenia	ZR	2 zad. · 12 pkt
5	**Kartkówka: metoda zachłanna, programowanie dynamiczne, podciągi – grupa A** 15 minut · 8 pkt · bez komputera	Kartkówka	ZR	4 zad. · 8 pkt 15 min
6	**Kartkówka: metoda zachłanna, programowanie dynamiczne, podciągi – grupa B** 15 minut · 8 pkt · bez komputera	Kartkówka	ZR	4 zad. · 8 pkt 15 min
7	**Sprawdzian: Podejście zachłanne i programowanie dynamiczne; podciągi – grupa A** 45 minut · 20 pkt · bez komputera	Sprawdzian	ZR	9 zad. · 20 pkt 45 min
8	**Sprawdzian: Podejście zachłanne i programowanie dynamiczne; podciągi – grupa B** 45 minut · 20 pkt · bez komputera	Sprawdzian	ZR	9 zad. · 20 pkt 45 min

Jak korzystać z zestawu

- Karta 1 przypomina najważniejsze wiadomości i pokazuje przykłady rozwiązane krok po kroku (z kodem w Pythonie albo opisem kroków w aplikacji) – od niej warto zacząć.
- Karta 2 to ćwiczenia podstawowe, karta 3 – utrwalenie w formule zadań maturalnych CKE (ramka „Jak to sprawdza CKE").
- Karta 4 to praca przy komputerze: pliki danych są w archiwum ZIP dołączonym do zestawu, a odpowiedzi kontrolne dla plików *_przyklad (pliki o nazwie z dopiskiem „_przyklad") podano przy zadaniach. Wyniki zapisuje się w plikach wynikiN.txt – jak na maturze.
- Karty 5–6 to kartkówka (15 minut), karty 7–8 – sprawdzian (45 minut), obie w grupach A i B: te same typy zadań i ta sama punktacja, inne dane. Kartkówka i sprawdzian nie wymagają komputera.
- Klucz zawiera odpowiedzi, przykładowe programy, zasady oceniania w stylu CKE i numery wymagań podstawy. Osobny plik „Powtórka" zbiera teorię, typowe błędy i test „Sprawdź się".

Oznaczenia

ZP	zakres podstawowy
ZR	zakres rozszerzony (treści tylko dla ZR)
(0–2 pkt)	liczba punktów możliwych do zdobycia za zadanie
Zadanie 4.1.	wiązka zadań do wspólnego materiału źródłowego
Jak to sprawdza CKE	typ zadania i typowy błąd ze sprawozdań CKE
Na maturze	wymagania z podstawy programowej, których dotyczy temat

Wymagania z podstawy programowej (Dz.U. 2024 poz. 1019)

- **I+II.R3d** objaśnia oraz porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przykłady problemów i algorytmów, w szczególności: d) podejście zachłanne (do wydawania reszty, szukania najkrótszej drogi),
- **I+II.R3e** objaśnia oraz porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przykłady problemów i algorytmów, w szczególności: e) programowanie dynamiczne (do szukania najdłuższego wspólnego podciągu),
- **I+II.R2c** wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie,

Materiał oryginalny – żadne zadanie nie jest kopią zadania CKE ani podręcznika; dane w plikach są fikcyjne. Podstawa programowa LO i technikum w brzmieniu z 2024 r. (Dz.U. 2024 poz. 1019). Wymagania: I+II.R3d, I+II.R3e, I+II.R2c. Miejsce w programach nauczania: „Informatyka na czasie" (Nowa Era) ZR – algorytmika (metody zachłanne, programowanie dynamiczne) (nazwy serii podano tylko informacyjnie). Do użytku własnego i pracy z własnymi uczniami.

Podejście zachłanne i programowanie dynamiczne; podciągi – najważniejsze wiadomości

Teoria, przykłady krok po kroku, rozgrzewka

ZR </> KARTA 1/8

Imię i nazwisko: Klasa: Data: Wynik: ..... / 7 pkt

Najważniejsze wiadomości · 7 pkt

Przypomnij sobie

- **Podejście zachłanne** (ang. *greedy*): rozwiązanie budujemy krok po kroku i w każdym kroku wybieramy to, co **w tej chwili** wygląda najlepiej, bez cofania decyzji. Algorytm jest szybki i prosty, ale nie zawsze daje rozwiązanie optymalne – trzeba to uzasadnić albo znaleźć kontrprzykład.
- **Wydawanie reszty zachłannie**: bierzemy największą monetę nie większą od pozostałej kwoty i powtarzamy. Dla polskich monet (1, 2, 5, 10, 20, 50 gr, 1, 2, 5 zł) daje to zawsze najmniejszą liczbę monet (taki system nominałów nazywamy kanonicznym). Dla nominałów 1, 3, 4 i kwoty 6 algorytm wyda 4 + 1 + 1 (3 monety), a wystarczą 3 + 3 (2 monety).
- **Najkrótsza droga – algorytm Dijkstry** (1959): odległość startu = 0, pozostałych = ∞. W każdym kroku **zachłannie** zatwierdzamy niezatwierdzony wierzchołek o najmniejszej odległości i poprawiamy odległości jego sąsiadów (d[v] = min(d[v], d[u] + waga)). Działa poprawnie tylko dla **nieujemnych wag** krawędzi.
- **Programowanie dynamiczne**: problem dzielimy na podproblemy, które się powtarzają; wynik każdego liczymy raz i zapisujemy w tablicy, od najmniejszych do największych (metoda wstępująca). Przykład – najmniejsza liczba monet dla dowolnych nominałów: ile[0] = 0, ile[k] = min(ile[k − m] + 1) po nominałach m ≤ k.
- **Najdłuższy wspólny podciąg** (NWP, ang. *LCS*) napisów a i b: podciąg powstaje przez skreślenie niektórych znaków (nie musi być spójny). T[i][j] – długość NWP prefiksów a[1..i] i b[1..j]: gdy a[i] = b[j], to T[i][j] = T[i−1][j−1] + 1, w przeciwnym razie T[i][j] = max(T[i−1][j], T[i][j−1]); T[i][0] = T[0][j] = 0. Tablica ma (m + 1)(n + 1) pól – czas rzędu m · n.
- **Najdłuższy spójny podciąg niemalejący** (fragment kolejnych wyrazów): jedno przejście – długość bieżącego fragmentu rośnie o 1, gdy a[i] ≥ a[i − 1], a w przeciwnym razie wraca do 1; zapamiętujemy największą długość. Czas rzędu n.
- **Spójny podciąg o największej sumie** (algorytm Kadane'a): s – największa suma fragmentu kończącego się na bieżącym wyrazie: s = max(x, s + x); wynik = największe s. Sprawdzanie wszystkich fragmentów wymaga rzędu n^2 sum (liczonych narastająco), algorytm Kadane'a – jednego przejścia (rzędu n).
- **Porównanie metod**: zachłanna – jedna decyzja na krok, szybka, poprawna tylko dla niektórych problemów; dynamiczna – rozważa wszystkie podproblemy, zawsze optymalna, ale potrzebuje tablicy (pamięci); przegląd zupełny – sprawdza wszystkie możliwości, zwykle za wolny dla dużych danych, ale przydaje się do testowania szybszych programów (wymaganie I.3).

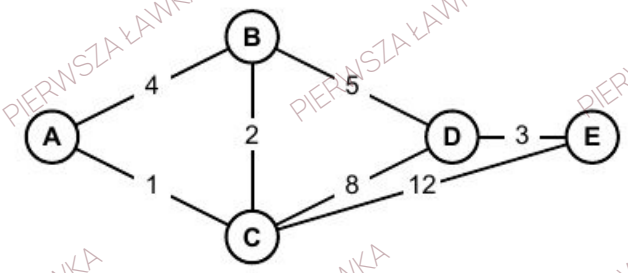
Przykłady z rozwiązaniem

Przykład 1. **Automat ma wydać 8,70 zł resztą. Jakie monety wyda algorytm zachłanny?**

```
def reszta(kwota, nominaly):
    # nominaly uporządkowane malejąco, np. [500, 200, 100, 50, 20, 10, 5, 2, 1]
    wydane = []
    for m in nominaly:
        while kwota >= m:
            wydane.append(m)
            kwota = kwota - m
    return wydane
```

Kwoty liczymy w groszach (unikamy błędów zaokrągleń liczb zmiennoprzecinkowych): 870 → 5 zł (zostaje 370) → 2 zł (170) → 1 zł (70) → 50 gr (20) → 20 gr (0). Razem 5 monet. Dla polskich nominałów to najmniejsza możliwa liczba monet.

Przykład 2. **Wyznacz najkrótsze drogi z A do pozostałych wierzchołków grafu (liczby przy krawędziach to długości dróg w km).**



Start: d(A) = 0, reszta ∞. Zatwierdzamy A → d(B) = 4, d(C) = 1. Zatwierdzamy C (najmniejsze d = 1) → d(B) = min(4, 1 + 2) = 3, d(D) = 1 + 8 = 9, d(E) = 1 + 12 = 13. Zatwierdzamy B (3) → d(D) = min(9, 3 + 5) = 8. Zatwierdzamy D (8) → d(E) = min(13, 8 + 3) = 11. Zatwierdzamy E. Wynik: B = 3, C = 1, D = 8, E = 11; kolejność zatwierdzania A, C, B, D, E.

PIERWSZA ŁAWKA

Przykład 3. **Oblicz długość najdłuższego wspólnego podciągu napisów KOTARA i TARAS.**

	–	T	A	R	A	S
–	0	0	0	0	0	0
K	0	0	0	0	0	0
O	0	0	0	0	0	0
T	0	1	1	1	1	1
A	0	1	2	2	2	2
R	0	1	2	3	3	3
A	0	1	2	3	4	4

Wiersze – litery napisu KOTARA, kolumny – litery napisu TARAS. Prawy dolny róg: **4**. Idąc od niego wstecz (po przekątnej przy równych literach) odczytujemy podciąg **TARA**.

```
def nwp(a, b):                          # długość najdłuższego wspólnego podciągu
    m, n = len(a), len(b)
    T = [[0] * (n + 1) for _ in range(m + 1)]
    for i in range(1, m + 1):
        for j in range(1, n + 1):
            if a[i - 1] == b[j - 1]:
                T[i][j] = T[i - 1][j - 1] + 1
            else:
                T[i][j] = max(T[i - 1][j], T[i][j - 1])
    return T[m][n]
```

Przykład 4. **Dla ciągu 3, −5, 4, −1, 2, −6, 5, 1 znajdź największą sumę spójnego fragmentu.**

i	x	s	naj
1	3	3	3
2	−5	−2	3
3	4	4	4
4	−1	3	4
5	2	5	5
6	−6	−1	5
7	5	5	5
8	1	6	6

Wynik 6 – fragment 5, 1 (dwa ostatnie wyrazy). Fragment 4, −1, 2 ma sumę 5.

```
def max_suma(a):
    s = a[0]      # suma najlepszego fragmentu kończącego się na x
    naj = a[0]    # najlepsza suma do tej pory
    for x in a[1:]:
        s = max(x, s + x)
        naj = max(naj, s)
    return naj
```

Podgląd z Pierwszej Ławki

Zadania

Zadanie 1. (0–2 pkt) ..... / 2

Ile monet wyda zachłannie automat z polskimi monetami?

1. 3,80 zł → _____ monet
2. 1,99 zł → _____ monet

Zadanie 2. (0–1 pkt) ..... / 1

Algorytm zachłanny w każdym kroku

A. wybiera możliwość najlepszą w danej chwili i nie zmienia wcześniejszych decyzji

B. sprawdza wszystkie możliwe rozwiązania i wybiera najlepsze

C. dzieli problem na dwie równe części

D. zapamiętuje w tablicy wyniki wszystkich podproblemów

Zadanie 3. (0–2 pkt) ..... / 2

Uzupełnij.

1. Najdłuższy spójny fragment niemalejący ciągu 5, 1, 1, 4, 2, 7, 7, 9 ma długość _____
2. Największa suma spójnego fragmentu ciągu −2, 6, −1, 3, −8, 4 to _____

Zadanie 4. (0–2 pkt) ..... / 2

Oceń prawdziwość zdań.

1. Podciąg napisu musi składać się z sąsiednich znaków tego napisu.	P	F
2. Algorytm Dijkstry w każdym kroku zatwierdza wierzchołek o najmniejszej bieżącej odległości.	P	F

Podejście zachłanne i programowanie dynamiczne; podciągi – ćwiczenia podstawowe

Ćwiczę umiejętności z działu

ZR </> KARTA 2/8

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 20 pkt

Ćwiczenia · 20 pkt

Zadanie 1. (0–3 pkt) ..... / 3

Automat wydaje zachłannie resztę 4,65 zł polskimi monetami (kwoty w groszach). Uzupełnij tabelę.

krok	kwota do wydania przed krokiem (gr)	wydana moneta (gr)
1	465	200
2		
3		
4		
5		

Zadanie 2. (0–3 pkt) ..... / 3

Automat do gry wydaje żetony o wartościach 1, 4 i 6. Kwota do wydania: 8.

1. Żetony wydane zachłannie: _____
2. Najmniejsza liczba żetonów: _____
3. Optymalny zestaw żetonów: _____

Zadanie 3. (0–3 pkt) ..... / 3

Programowanie dynamiczne: ile[k] to najmniejsza liczba żetonów (1, 4, 6) dla kwoty k; ile[k] = min(ile[k − m] + 1). Uzupełnij tabelę.

k	0	1	2	3	4	5	6	7	8	9	10
ile[k]	0	1	2	3							

Zadanie 4. (0–2 pkt) ..... / 2

Napisy BRAMA i RAMKA.

1. Długość najdłuższego wspólnego podciągu: _____
2. Przykładowy najdłuższy wspólny podciąg: _____

Zadanie 5. (0–3 pkt) ..... / 3

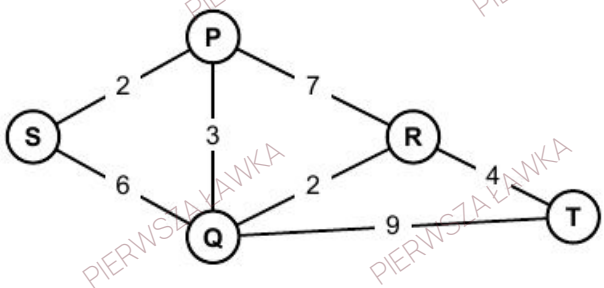
Wykonaj funkcję max_suma (algorytm Kadane'a) dla ciągu 2, −4, 3, −1, 5, −7, 4. Uzupełnij tabelę.

```
def max_suma(a):
    s = a[0]      # suma najlepszego fragmentu kończącego się na x
    naj = a[0]    # najlepsza suma do tej pory
    for x in a[1:]:
        s = max(x, s + x)
        naj = max(naj, s)
    return naj
```

i	x	s	naj
1	2	2	2
2	−4	−2	2
3	3		
4	−1		
5	5		
6	−7		
7	4		

Zadanie 6. (0–4 pkt) ..... / 4

Algorytmem Dijkstry wyznacz najkrótsze odległości od wierzchołka S.



1. d(P) = _____ 2. d(Q) = _____ 3. d(R) = _____ 4. d(T) = _____

Zadanie 7. (0–2 pkt) ..... / 2

Uzupełnij luki w funkcji, która zwraca największą sumę spójnego fragmentu listy a.

```
def max_suma(a):
    s = a[0]      # suma najlepszego fragmentu kończącego się na x
    naj = a[0]    # najlepsza suma do tej pory
    for x in a[1:]:
        s = max(x, luka 1)
        naj = max(luka 2)
    return naj
```

1. luka 1: _____
2. luka 2: _____

Zadanie 4. (0–3 pkt) ..... / 3

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który wyznaczy długość najdłuższego spójnego podciągu niemalejącego (fragmentu złożonego z kolejnych wyrazów) ciągu A.

Przykład: dla A = [2, 2, 5, 3, 3, 4, 6, 1] wynikiem jest 4 (3, 3, 4, 6).

Uwaga: Nie używaj funkcji wbudowanych (np. max) ani dodatkowych tablic.

Specyfikacja

Dane: n – liczba całkowita, $n \geq 1$; A[1..n] – tablica liczb całkowitych

Wynik: naj – długość najdłuższego fragmentu A[i..j], w którym $A[i] \leq A[i + 1] \leq \ldots \leq A[j]$

Miejsce na odpowiedź

Zadanie 5. (0–2 pkt) ..... / 2

Automat wydaje żetony o wartościach 1, 7 i 10. Podaj najmniejszą kwotę, dla której algorytm zachłanny nie wyda najmniejszej liczby żetonów, oraz oba zestawy: zachłanny i optymalny.

Miejsce na odpowiedź

Zadanie 6. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.

Obliczenie długości najdłuższego wspólnego podciągu napisów o długościach m i n metodą programowania dynamicznego (tablica T) wymaga liczby operacji rzędu

A. $m + n$ B. 2^{m+n} C. $m \cdot n$ D. $\log(m \cdot n)$

Zadanie 7. (0–2 pkt) ..... / 2

Podaj największą sumę spójnego fragmentu ciągu 4, −6, 3, 5, −2, 6, −9, 2 oraz ten fragment.

Miejsce na odpowiedź

CO JESZCZE JEST W ŚRODKU

Podejście zachłanne i programowanie dynamiczne; podciągi – kartkówka, sprawdzian i karty pracy z powtórką, informatyka, klasa 4 LO i technikum...

Podgląd pokazał 9 z 28 stron. Pełna wersja: 28 stron A4, PDF do druku i DOCX do edycji.

- Karta 1: Podejście zachłanne i programowanie dynamiczne; podciągi – najważniejsze wiadomości – 4 zadania, 7 pkt
- Karta 2: Podejście zachłanne i programowanie dynamiczne; podciągi – ćwiczenia podstawowe – 7 zadań, 20 pkt
- Karta 3: Podejście zachłanne i programowanie dynamiczne; podciągi – utrwalenie – 7 zadań, 13 pkt
- Karta 4: Podejście zachłanne i programowanie dynamiczne; podciągi – w praktyce, przy komputerze – 2 zadania, 12 pkt
- Karta 5: Kartkówka: metoda zachłanna, programowanie dynamiczne, podciągi – grupa A – 4 zadania, 8 pkt
- Karta 6: Kartkówka: metoda zachłanna, programowanie dynamiczne, podciągi – grupa B – 4 zadania, 8 pkt
- Karta 7: Sprawdzian: Podejście zachłanne i programowanie dynamiczne; podciągi – grupa A – 9 zadań, 20 pkt
- Karta 8: Sprawdzian: Podejście zachłanne i programowanie dynamiczne; podciągi – grupa B – 9 zadań, 20 pkt Karta 1 przypomina teorię i pokazuje przykłady rozwiązane krok po kroku, d

Kup pełną wersję – plik od razu na e-mail

pierwszalawka.pl/material/informatyka-liceum-klasa-4-zr-dzial-11-podejscie-zachlanne-i-programowanie-dynamiczne