Zajęcia 17

Temat: Programowanie zachłanne, dynamiczne 2

Czas trwania: 2x45 min

Cel zajęć:

projektuje i programuje proste problemy z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, tablice, rekurencję, pisze własne funkcje rekurencyjne, przetwarza tablice dwuwymiarowe, algorytmy zachłanne testuje poprawność programów dla różnych danych, posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

Efekty:

- umie uruchomić potrzebne oprogramowanie,
- umie napisać program z wykorzystaniem funkcji rekurencyjnej, równaniami rekurencyjnymi,
- zna operacje na zbiorach,
- zna przeszukiwanie z powrotami

Formy i metody pracy: praca samodzielna,

Zadania do wykonania na zajęciach	Treści programowe
1. Poszukiwania (szu)	M.3, P.2.17, P.2.19, A.3.5
2. Stadion	M.3, P.2.17, P.2.19, A.3.5
3. Zbiórka	M.3, P.2.17, P.2.19, A.3.5

Materiały do zajęć:

https://www.main2.edu.pl/main2/courses/show/7/20/

https://www.main2.edu.pl/main2/courses/show/7/22/

Zadania do wykonania w domu:

Bracia (V OIG):

https://szkopul.edu.pl/problemset/problem/KkZDP16mELVF692Bf-NJsN1L/site/

Monety (V OIG):

https://szkopul.edu.pl/problemset/problem/GMBfzRQAdw3b7wfvC9Xf7YE4/site/

ZADANIA I ROZWIĄZANIA

Zadanie 1. Poszukiwania

Dostępna pamięć: 32MB

Książe Bratumił II Wstydliwy przez stosunek do białogłów swój przydomek otrzymał. Przeto jego rodziciel, król Rufin IV Rzadkowąsy, martwił się wielce. 40 lat Bratumiła dojrzałością późną na pewno zwać nie można, wszelako żona i dzieci dla syna chęcią przedłużenia rodu królewskiego nieodzowne Rufinowi wydawać się mogły. Toteż synowi nakazał, by królestwo całe przewędrował i za żonę jakową dorodną infantkę pojął. Syn wolę ojca spełnić zaprzysiągł, aliści warunek także wysunął: spędzi w podróży na poszukiwaniach księżniczki o dwa dni mniej, niż monarchia ojca hrabstw w szerokości i w długości w sumie liczy, a we wszelkim odwiedzonym hrabstwie dokładnie jeden dzionek pobędzie. Rufin tą przymusowością strapił się bardzo, azaliż synowi przyrzeczenie dał. Wszelako sam mu szlak zdeklarował się wyznaczyć taki, ażeby Bratumił na jak największą liczbę księżniczek okiem rzucił.

Wtórą noc już Rufin oka zmrużyć nie może. Jak trasę wyznaczyć? Na szczęście informatyk z eskapady dalekiej akuratnie przybył i laptop najnowszej generacji z procesorem wielordzeniowym sprowadził. Tedy król pchnął umyślnego po nadwornego, by mu ową trasę ustalił.

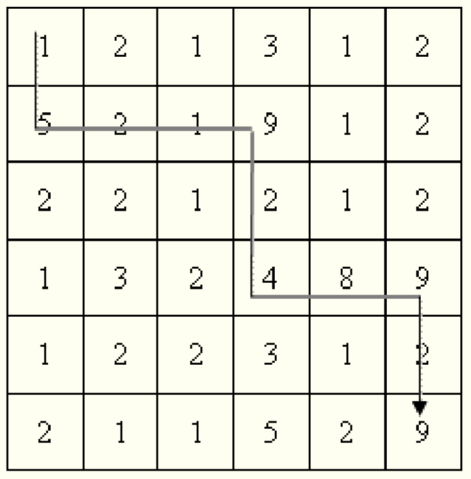
Zadanie

Mapa państwa to tablica kwadratowa o wymiarach n x n. Każdy kwadrat określa jedno hrabstwo, w każdej komórce znajduje się liczba zamieszkujących go księżniczek.

Napisz program, który dla danej tablicy liczb n x n oblicza ścieżkę o maksymalnej liczbie księżniczek prowadzącą od lewego górnego pola tablicy do prawego dolnego. Ścieżka składa się z sekwencji kroków. Liczba kroków wynosi dokładnie $2 \cdot n - 2$.

Maksymalną liczbą księżniczek ścieżki jest suma liczb w każdym z odwiedzonych pól tablicy.

Maksymalna ścieżka w przykładowej tablicy o wymiarze 6 x 6 pokazana jest poniżej.



Wejście

W pierwszym wierszu wejścia zapisana została jedna liczba całkowita n, oznaczającą liczbę wierszy oraz kolumn tablicy. Dalej następuje n x n liczb naturalnych nie większych niż 1000, w porządku wiersz-kolumna, tj. pierwszych n liczb tworzy pierwszy wiersz tablicy, następnych n liczb tworzy drugi wiersz itd. Liczby w wierszu są od siebie oddzielone przynajmniej jedną spacją. Liczba wierszy i kolumn jest z zakresu od 1 do 1000 włącznie.

Wyjście

Pierwsza i jedyna liczba określa maksymalną liczbę księżniczek, jakie może poznać Bratumił.

Przykład

Wejście	Wyjście
3 1 1 1 1 1 1 1 1 1	5
Wejście	Wyjście
6 1 2 1 3 1 2 5 2 1 9 1 2 2 2 1 2 1 2 1 3 2 4 8 9 1 2 2 3 1 2 2 1 1 5 2 9	52

Rozwiązanie

Obserwacja: Zauważmy, że Bratumił może się poruszać po planszy wyłącznie w prawo lub w dół (w innym wypadku liczba odwiedzonych hrabstw byłaby zbyt duża).

Rozwiązanie wolne: Możliwe jest generowanie wszystkich możliwych do uzyskania ścieżek i wyznaczanie dla nich sumy.

Zadanie dla uczniów: ile różnych ścieżek jest możliwych do wygenerowania dla tablicy n x n?

Rozwiązanie szybkie: Korzystając z obserwacji podanej na wstępie można zauważyć, że do każdego hrabstwa w pierwszym wierszu możemy przyjść wyłącznie z lewej strony. Podobnie do każdego hrabstwa w pierwszej kolumnie możemy przyjść wyłącznie z góry. Pozwala to w prosty sposób wyznaczyć liczbę księżniczek dla każdego pola w pierwszym wierszu i w pierwszej kolumnie. Idąc tym tokiem myślenia można zauważyć, że do hrabstwa o współrzędnych (2,2) możemy przyjść wyłącznie z hrabstw (2,1) lub (1,2). Oczywiście aby zmaksymalizować liczbę księżniczek będziemy wybierać to hrabstwo, które oferuje nam wyznaczoną dotychczas większą sumę. Podobnie postąpimy dla kolejnych pól w wierszu nr 2, a następnie kolejno w pozostałych wierszach.

Deklarując tablicę z pustym wierszem i kolumną o indeksie 0 (z wartościami 0 dla tych pól) nasz algorytm pozostanie stały bez dodatkowego wyznaczania wartości dla pierwszego wiersza i pierwszej kolumny.

```
Tablica a[0..n]
wczytaj n
dla y=1,2,…,n wykonaj
     dla x=1,2,…,n wykonaj
```

```
          wczytaj a[x,y]
dla y=1,2,…,n wykonaj
     dla x=1,2,…,n wykonaj
          a[x,y] ← maksimum(a[x-1,y],a[x,y-1]) + a[x,y]
wypisz a[x,y]
```

Zadanie dla uczniów: Jak odtworzyć ścieżkę, po której poruszał się Bratumił?

Zadanie 3. Zbiórka

Limit pamięci: 64MB

Pałac króla Bitolandii jest w ruinie, a kasa państwa świeci pustkami! Obywatele Bitolandii postanowili przeprowadzić zbiórkę pieniędzy na ratowanie cennego zabytku. W centrum stolicy, Bitogrodu, ustawiono wielką skrzynię, do której każdy może wrzucić datek. Zbiórka trwa już jakiś czas. Nikt jednak nie wie, jaka kwota mogła się dotychczas uzbierać. Gdyby tyko społeczny komitet ratowania zabytku miał pewność, że zebrane środki są wystarczające, zakończono by kwestę i przystąpiono do prac. W Bitolandii używa się tylko monet, a każda ma określoną wagę i nominał. Czy znając wagę zebranych monet możliwe jest określenie minimalnej kwoty, jaką do tej pory zebrano?

Wejście

Pierwszy wiersz zawiera jedną liczbę całkowitą c – ciężar zgromadzonych pieniędzy, nie więcej niż 10000. W drugiej linii znajduje się jedna liczba całkowita n – liczba różnych monet używanych w danej walucie ($1 \le n \le 1000$). W kolejnych liniach znajdują się opisy monet zawierające po dwie liczby całkowite, nominał m oraz wagę w ($1 \le m \le 50000$, $1 \le w \le 10000$).

Wyjście

Wypisz minimalną ilość pieniędzy, które można osiągnąć za pomocą monet z danej masy całkowitej. Jeśli waga nie może być osiągnięta, należy wypisać komunikat "NIEMOZLIWE".

Przykład

Wejście:	Wyjście:
100 2 1 10 20 5	10

Wejście:	Wyjście:
100 2 10 1 20 5	400

Wejście:	Wyjście:
5 2 2 2 4 4	NIEMOZLIWE

Rozwiązanie

Zauważmy, że rozwiązanie zadania to klasyczny problem plecakowy, jednak zamiast szukać wartości maksymalnej kwoty dla podanej wagi, szukamy wartości minimalnej.

W tablicach `waga_monet[]` oraz `wartość_monet[]` zapamiętajmy odpowiednio wagi i wartości poszczególnych monet (nie więcej niż 1000). Zauważmy, że maksymalna waga wyniesie 10000, potrzebujemy więc jeszcze tablicy `najlepsza[]` przechowywujące najmniejsze wartości dla każdej możliwej do uzyskania wagi.

Najlepsza możliwa wartość do uzyskania wagi 0 wynosi oczywiście 0. Teraz dla każdej wagi aż do wagi całej zawartości skrzyni będziemy sprawdzali, czy biorąc po kolei każdą monetę o określonej wadze mogę uzyskać lepszą (mniejszą) kwotę.

Na koniec sprawdzamy, czy w `najlepsza[szukana_waga]` została przez nas znaleziona jakaś waga.

```
tablica waga_monet[1..1000]
tablica wartość_monet[1..10000]
tablica najlepsza[1.1000]
wczytaj c, n
dla i=1,2,…,n
  wczytaj wartość_monet[i], waga_monet[i]
najlepsza[0] ← 0
dla j=1,2,…,c
  najlepsza[j] ← ∝
    dla i=1,2,…,n
      jeżeli waga_monet[i] ≤ j
        jeżeli najlepsza[j-waga[monet[i]]+wartość_monet[i]<najlepsza[j]
          najlepsza[j] ← najlepsza[j-waga[monet[i]]+wartość_monet[i]
jeżeli najlepsza[c] < ∝
  wypisz najlepsza[c]
przeciwnie
  wypisz „NIEMOŻLIWE”
```