Rekurencja i dekompozycja

Maciej Hojda

1 Zadania

1.1 Ciągi liczbowe

Dla następujących ciągów liczbowych

1. $\forall n \in \{1, 2, \ldots\} : x(n) = 3^n + x(n-1)$
   $x(0) = 0$
2. $\forall n \in \{1, 2, \ldots\} : x(n) = n + x(n-2)$
   $x(-1) = x(0) = 0$
3. $\forall n \in \{2, 3 \ldots\} : x(n) = x(n-1) + x(n-2)$
   $x(1) = 1; x(0) = 0$

wykonaj:

1. zaimplementuj (rekurenycjny) algorytm wyliczający wartość $n$tego elementu ciągu,
2. analitycznie wyznacz wzór na wartość $n$tego elementu ciągu (indukcja),
3. napisz procedurę weryfikującą poprawność zaimplementowanej rekurencji (wyświetlającą i porównującą wynik numeryczny i analityczny) dla N pierwszych elementów ciągu.

1.2 Grafy

Dany jest graf nieskierowany $\mathbb{V} = (\mathbf{V}, \mathbf{E})$, gdzie $\mathbf{V} = \{v_1, v_2, \ldots, v_V\}$ i $\mathbf{E} = \{e_1, e_2, \ldots, e_E\}$, gdzie $e_i = \{j, k\} : j, k \in \mathbf{V}$. Napisz program który działa jak następuje.

1. Dla każdego wierzchołka losuje pozycję (v_i^x, v_i^y) na płaszczyźnie $[0, 100] \to [0, 100]$.
2. Dla zadanego V generuje drzewo rozpinające wierzchołki $\mathbf{V}$ w następujący sposób:
   (a) do wyczerpania wierzchołków, wybieraj kolejny wierzchołek v z $\mathbf{V}$,
   (b) do grafu $\mathbb{V}$ dodaj krawędź $\{v, w\}$, gdzie w jest najbliższym (odległość euklidesowa) wierzchołkiem, który jest już połączony z v_1 jakąkolwiek ścieżką (w szczególności może to być ścieżka trywialna, czyli $w = v_1$).
3. Wyświetla generację drzewa krok po kroku (np. co sekundę), zaczynając od (kolejno ponumerowanych) wierzchołków bez połączeń, i dodając połączenia jedno za drugim. Na połączeniach wyświetlana jest odległość zaokrąglona do 2 miejsc po przecinku.

Uwaga: Upewnij się, że przy każdym uruchomieniu programu dla takich samych V, program zawsze generuje to samo drzewo – wykorzystaj `random.seed()`.

1.3 Odległość

Dla zadanego grafu $\mathbb{V} = (\mathbf{V}, \mathbf{E})$, którego wierzchołki $v \in \mathbf{V}$ mają ustaloną pozycję (v_i^x, v_i^y) na płaszczyźnie, napisz dwuargumentową funkcję $\texttt{dist}(v, w)$, która zwraca odległość między zadanymi w argumentach wierzchołkami v, w (mogą być numery wierzchołków). Funkcja działa jak następuje.

1. Jeśli w jest bezpośrednim sąsiadem v, to odległość jest równa odległości euklidesowej.
2. Jeśli w nie jest bezpośrednim sąsiadem v, to dla każdego bezpośredniego sąsiada s wierzchołka v uruchom funkcję $\texttt{dist}(s, w)$ na grafie ze zbiorem wierzchołków pomniejszonym o v. Ustal odległość na $\min_{s \in \mathbf{S}}\{\texttt{dist}(s, w) + d_{v,s}\}$ gdzie $\mathbf{S}$ to zbiór bezpośrednich sąsiadów wierzchołka v, a $d_{v,s}$ to odległość euklidesowa między wierzchołkami v, s.

Rozbuduj funkcję tak, aby zwracała też ścieżkę, która odpowiada wyliczonej odległości.

1.4 Kolorowanie

Centrach handlowe (galerie) pewnego miasta zostały zmonopolizowane tak, że zarządza nimi jeden właściciel. W każdej galerii realizowany jest szereg usług podstawowych. Aby zróżnicować usługi dostępne w każdej galerii, a tym samym, aby rozdystrybuować klientów między wieloma galeriami, każda z nich świadczy jedną usługę dodatkową. Zbiór usług dodatkowych jest ustalony i ma on liczność L. Należy zagwarantować, aby żadna sąsiadująca ze sobą galeria nie świadczyła tej samej usługi dodatkowej – w przeciwnym wypadku mamy konflikt usług i rozwiązanie jest niedopuszczalne.

Zaproponuj strukturę danych, która będzie przechowywać informacje o centrach handlowych rozmieszczonych w mieście.

Zaimplementuj algorytm rozmieszczania usług dodatkowych $\texttt{usl}(A, B)$, który działa jak nastepuje:

- wybiera jedną galerię g ze zbioru A (zbiór galerii bez przydzielonych usług) i przydziela do niej jedną usługę,
- jeśli na zbiorze galerii B powstaje konflikt usług, to wybierana jest inna usługa; jeśli nie można przydzielić usługi tak, aby nie doszło do konfliktu, to zwracana jest informacja o niedopuszczalności rozwiązania,
- wywoływany jest algorytm z pomniejszonym zbiorem galerii $\texttt{usl}(A - \{g\}, B)$; jeśli uruchomiona podrzędna instancja algorytmu zwróciła rozwiązanie dopuszczalne, to jest ono zwracane przez algorytm, razem usługą przydzieloną w pierwszym punkcie.

Algorytm jest uruchamiany wywołaniem $\texttt{usl}(A, A)$, gdzie A to zbiór wszystkich galerii.

1.5 Rozmieszczenie

Na interesujący geologicznie obszar została nałożona (wirtualna) siatka potencjalnych lokalizacji odwiertów. Siatka ma rozmiary $N \times K$, gdzie odległość między kolejnymi punktami siatki w pionie i w poziomie to D metrów. Ze względu na poglądowy charakter tych badań geologicznych jest istotne, aby odwierty nie odbywały się zbyt gęsto, co pozwoli na uniknięcie nadmiernego duplikowania rezultatów. Konkretnie, warto wykonywać odwierty w punktach siatki odległych od siebie o przynajmniej S metrów. Z drugiej strony, wykonanie odwiertów jest kosztowne, a ich maksymalna liczba przewidziana w budżecie inwestycji to R.

Napisz program, który dla ustalonych N, K, D, S, R wyznaczy plan prac geologicznych tak, aby maksymalizować liczbę faktycznie zaplanowanych odwiertów przy zachowaniu wspomnianych ograniczeń. Program uruchom dla przykładowych danych, a rezultat przedstaw w sposób graficzny (generowany automatycznie, w programie).