

Projekt I

Algorytmy generowania podzbiorów i permutacji.

Algorytm generowania podzbiorów zbioru $\{1, \dots, n\}$.

- pierwszy podzbiór to \emptyset ;
- kolejny podzbiór po podzbiorze A :
 - znajdujemy największy element nie należący do A , czyli $a = \max\{i \notin A\}$;
 - jeżeli nie ma takiego a , to rozważany podzbiór A jest ostatnim - Koniec;
 - w przeciwnym przypadku, dodajemy a do zbioru A i usuwamy z A wszystkie elementy większe od a .

Algorytm generowania k -elementowych podzbiorów zbioru $\{1, \dots, n\}$.

- pierwszy podzbiór to $\{1, \dots, k\}$;
- kolejny podzbiór po podzbiorze $A = \{a_1, \dots, a_k\}$, gdzie $a_1 < \dots < a_k$:
 - znajdujemy najmniejsze i takie, że $a_i + 1 \notin A$;
 - jeżeli $a_i = a_n$, to rozważany podzbiór $A = \{n - k + 1, \dots, n\}$ jest ostatnim - Koniec;
 - w przeciwnym przypadku, zwiększamy a_i o jeden, a elementy mniejsze od a_i zamieniamy na $i - 1$ najmniejszych kolejnych liczb, tzn. $a_j := j$, dla $j < i$.

Algorytm generowania permutacji zbioru $\{1, \dots, n\}$.

- pierwsza permutacja to $a_i = i$, dla $1 \leq i \leq n$,
- kolejna permutacja po permutacji $(a_1 \dots a_n)$:
 - znajdujemy największe j spełniające warunek $a_j < a_{j+1}$
 - jeżeli nie ma takiego j , to rozważana permutacja jest permutacją ostatnią - Koniec;
 - w przeciwnym przypadku, zamieniamy a_j z najmniejszym a_k takim, że $a_k > a_j$ i $k > j$, a następnie odwracamy porządek elementów a_{j+1}, \dots, a_n

ZADANIA (8 pkt)

Zad. 1. (1 pkt) Zaimplementować algorytm generowania podzbiorów zbioru $\{1, \dots, n\}$. Wypisz 10 kolejnych podzbiorów zbioru $\{1, 2, 3, 4, 5, 6, 7\}$ poczynając od podzbioru $\{1, 2, 3, 5\}$.

Założenia:

- funkcja generująca podzbiory zbioru głównego $1, \dots, n$ zaczynając od zbioru X ma być postaci `function [A] = podzbior(n,X)`, gdzie
 - A - kolejne generowane podzbiory
 - n - największy element zbioru głównego
 - X - pierwszy zbiór, od którego generowane są pozostałe zbiory
- Elementy wygenerowanych podzbiorów mogą być wyświetlane w formie macierzy (niektóre elementy będą zerowe).

Zad. 2. (1 pkt) Zaimplementować algorytm generowania k -elementowych podzbiorów zbioru $\{1, \dots, n\}$. Wypisz 10 kolejnych 5-elementowych podzbiorów zbioru $\{1, 2, 3, 4, 5, 6, 7\}$.

Założenia:

- funkcja generująca p k -elementowych podzbiorów zbioru głównego $1, \dots, n$ ma być postaci `function kelem_podzbior(n,p,k)`, gdzie
 - n - największy element zbioru głównego
 - p - ilość generowanych podzbiorów
 - k - ilość elementów w podzbiorze
- Elementy wygenerowanych podzbiorów mogą być wyświetlane w formie macierzy $p \times k$

Zad. 3. (1 pkt) Zaimplementować algorytm generowania permutacji zbioru $\{1, \dots, n\}$. Wypisz 10 kolejnych permutacji zbioru $\{1, 2, 3, 4, 5, 6\}$ poczynając od permutacji (456321).

Założenia:

- funkcja generująca p permutacji zbioru głównego $1, \dots, n$ zaczynając od zadanej permutacji X ma być postaci `function permutacje(n,p,X)`, gdzie
 - n - największy element zbioru głównego
 - p - ilość generowanych permutacji
 - X - początkowa permutacja
- Elementy wygenerowanych podzbiorów mogą być wyświetlane w formie macierzy

Zad. 4. (5 pkt) Przedstawić przykład zastosowania wybranego z 3 zaimplementowanych algorytmów.

- Przedstawić krótki opis zagadnienia.
- Przedstawić motywację wyboru algorytmu do rozwiązania wybranego zagadnienia.
- Przedstawić wyniki - praca nad czytelną wizualizacją.

Projekt I składa się z 3 etapów:

- Rozdanie i omówienie wymagań odnośnie projektu: **30.10.2023**
- Implementacja algorytmów generowania: **13.11.2023**
- Wybór praktycznego zastosowania wybranego algorytmu generowania: **13.11.2023**
- Prezentacja wyników i ocena punktowa: **4.12.2023**

Pliki z rozwiązaniami (sprawozdanie w .pdf) oraz kodami (kod również należy umieścić w sprawozdaniu) należy przesłać jako niespakowane załączniki jednym listem elektronicznym o temacie **PROJEKT1_MD23Z** najpóźniej do **1.12.2023**. na adres:

zofia.grudziak@pw.edu.pl

W treści listu należy podać imiona i nazwiska osób wykonujących projekt.