

ASD – klasówka2
12.01.2026 r.

Zadanie 1 [7 punktów]

- a) Podaj permutację $P = \langle p_1, p_2, \dots, p_n \rangle$ liczb całkowitych od 1 do n taką, że po wstawieniu do początkowo pustego drzewa typu „splay” kolejno elementów permutacji P dostaniemy pełne drzewo binarne.
- a1) [1 punkt] Dla $n = 3$.
- a2) [3 punkty] Dla $n = 7$.
- a3) [bonus] Czy potrafisz wyznaczyć taką permutację dla dowolnego $n = 2^h - 1$, gdzie h jest liczbą całkowitą większą od 0.
- b) [3 punkty] Udowodnij, że dla dowolnego n -wierzchołkowego drzewa typu „splay”, istnieje ciąg zapytań o elementy drzewa (operacje search z wynikiem pozytywnym), po wykonaniu których dostaniemy drzewo BST o wysokości $n-1$.

Uwaga: w tym zadaniu przyjmujemy implementację wstawiania do drzewa „splay”, w której najpierw element jest wstawiany a dopiero potem na wstawionym elemencie jest wykonywana operacja „splay”.

Zadanie 2 [8 punktów]

Zaprojektuj strukturę danych dla dynamicznie zmieniającego się ciągu dodatnich liczb całkowitych $C = [c_1, c_2, \dots, c_n]$ umożliwiającą efektywne wykonywanie następujących operacji:

Ini(C):: $C := \langle \rangle$; //inicjalizacja pustego ciągu, wykonywana tylko raz na samym początku

Insert(C, x, i):: wstaw x na pozycję i w ciągu C ; //elementy, które dotychczas były na pozycjach $i, i+1, \dots, n$ będą teraz na pozycjach $i+1, i+2, \dots, n+1$

Delete(C, i):: usuń i -ty element z ciągu C ; //elementy, które dotychczas były na pozycjach $i+1, i+2, \dots, n$ będą teraz na pozycjach $i, i+1, \dots, n-1$

Search(C, i):: podaj wartość i -tego elementu w ciągu C ;

Sum2(i, j):: podaj największą sumę dwóch elementów z różnych pozycji w podciągu $C[i..j]$, $1 \leq i < j \leq n$;

By3(C):: podaj dla ilu elementów w ciągu C wartość wyrażenia $c_i + i$ jest podzielna przez 3.

Uwaga: możesz założyć, że podane operacje są zawsze wykonywalne dla zadanych parametrów.

Zadanie 3 [5 punktów]

Na polach nieskończonej, liniowej planszy układamy żetony. Pola są ponumerowane kolejnymi liczbami całkowitymi licząc od zera. Początkowo wszystkie pola są puste. Startujemy od pola o numerze 0. Każdy kolejny żeton możemy dołożyć do dowolnego niepustego stosu żetonów na jednym z pól lub położyć na pierwszym (o najmniejszym numerze) pustym polu. Jeśli stos, na który położyliśmy żeton osiągnął wysokość $h > 1$, taką samą jak wysokość stosu żetonów na którymkolwiek sąsiednim polu, to przekładamy ostatnio położony żeton na pierwsze wolne pole a kolejne $h-2$ żetonów z tego stosu kładziemy na kolejne wolne pola w taki sposób, że na każdym z nich znajdzie się liczba żetonów będąca potęgą liczby 2 i że żadne dwie potęgi 2 się nie powtórzą. Możesz przyjąć, że powstające stosy mają wysokości od najmniejszej do największej. Za operacje elementarne przyjmujemy zdjęcie żetonu z pola na planszy oraz położenie żetonu na pole na planszy. Podaj koszt amortyzowany dołożenia nowego żetonu na planszę.

Uzasadnij poprawność swoich rozwiązań i dokonaj analizy złożoności obliczeniowej zaproponowanych algorytmów.