

EGZAMIN Z AISD. NR 1.

IIUWr. II rok informatyki.

Numer indeksu:

1. W trakcie egzaminu miałeś za zadanie napisać algorytm Karatsuby. Zdaje się, że lekko się pomyliłeś i napisałeś:

```

multiply(a, b)
  n ← max(|a|, |b|)  (* |x| oznacza długość liczby x *)
  if n jest małe then pomnóż a i b klasycznym algorytmem
                        return obliczony iloczyn

  p ← ⌊n/2⌋
  a1 ← ⌊a/2p⌋; a0 ← a mod 2p
  b1 ← ⌊b/2p⌋; b0 ← b mod 2p
  z ← multiply(a0, b0)
  y ← multiply(2a1 + a0, 2b1 + b0)
  x ← multiply(a1, b1);
  return

```

Dokończ instrukcję **return** tak, by algorytm miał sens i robił to co powinien (i to tak szybko, jak powinien).

2. Jak już zapewne dobrze wiesz po ostatniej kartkówce, do obliczania elementów pewnych ciągów zdefiniowanych rekurencyjnie można zastosować metodę szybkiego podnoszenia macierzy do potęgi, a następnie wynik należy przemnożyć przez pewien wektor. Ułóż odpowiednią macierz dla elementów ciągu zdefiniowanego w następujący sposób:

$$h_{n+1} = \begin{cases} 7 & \text{dla } n = 1 \\ h_n + n^3 & \text{dla } n > 1 \end{cases}$$

- Do której potęgi należy podnieść tę macierz?
- Przez jaki wektor należy ją przemnożyć?
- Skąd odczytamy wynik (tj. wartość h_n)?

- 2

5. Podaj kolejność, w jakiej podzbiory będą dołączane do rozwiązania przez podany na wykładzie zachłanny algorytm dla problemu Pokrycia Zbioru (Set Cover) dla następujących danych:

- uniwersum $U = \{e_1, e_2, e_3, e_4, e_5\}$,
- podzbiory z ich wagami:
 - $S_1 = \{e_1, e_2, e_3\}; \quad w(S_1) = 9$
 - $S_2 = \{e_1, e_2, e_3, e_4, e_5\}; \quad w(S_2) = 12$
 - $S_3 = \{e_4, e_5\}; \quad w(S_3) = 5$
 - $S_4 = \{e_1, e_3, e_5\}; \quad w(S_4) = 6$

Odpowiedź uzasadnij.

6. Na wykładzie rozważaliśmy dwie wersje wykonania operacji deletemin na kopcu. W jednej wersji usuwany element zastępowaliśmy elementem ze skrajnie prawego liścia ostatniego poziomego kopca; w drugiej wersji przesuwaliśmy dziurę powstałą po usunięciu minimum na spód kopca. W której wersji spodziewaliśmy się wykonać mniej porównań kluczy i dlaczego?

7. Napisz, które z poniższych algorytmów sortują dane w sposób stabilny, a które nie. Dla tych drugich podaj przykład danych, które mogą zostać posortowane niestabilnie (w każdym przypadku uzasadnij ten fakt).
- InsertSort
 - MergeSort
 - Quicksort
 - HeapSort
 - SelectSort