AiSD wszystkie - wersja z dnia 16 lipca 2017

Zadania z części 1.

Drzewa AVL, B-drzewa

1. (Zad. 12, cz. 1, 06.2017) Jak mocno można ograniczyć (w pesymistycznym przypadku) liczbę rotacji podczas usuwania wierzchołka z drzewa AVL o n wierzchołkach? Uzasadnij, że nie da się bardziej niż podałeś(aś).

2. (Zad. 19, cz. 1, 06.2016) Jaką największą wysokość może mieć drzewo AVL zawierające 67 kluczy? Odpowiedź uzasadnij.

   Rozwiązanie:

   Zbudujmy drzewo AVL o maksymalnej wysokości przy użyciu jak najmniejszej ilości wierzchołków. Nazwijmy je T_i Dla $T_0 = 0$ będzie to drzewo puste, dla $T_1 = 1$ będzie to drzewo jednoelementowe, dla $T_2 = 2$ będzie to drzewo składające się z korzenia i jednego syna. Dla T_n, gdzie $n \geq 3$ będzie to drzewo które składa się z korzenia i dwóch poddrzew, którymi są drzewa o wysokości T_{n-1} oraz T_{n-2}. Drzewa te nazywamy drzewami Fibbonacciego. Liczba wierzchołków takiego drzewa to $N_h = N_{h-1}+1+N_{h-2}$. W takim razieiczba wierzchołków kolejnych drzew Fibbonacciego to odpowiednio $0, 1, 2, 4, 7, 12, 20, 33, 54, 88, \ldots$. Drzewo o 67 kluczach będzie miało więc maksymalnie wysokość 8, ponieważ, żeby zbudować drzewo AVL o wysokości 9 potrzeba minimum 88 kluczy.

   ---

3. (Zad. 14, cz. 1, 06.2015) Jeśli w drzewach AVL zmienilibyśmy warunek, by poddrzewa mogły różnić się o 2 (nie o 1) wysokością, to Czy drzewo n-wierzchołkowe dalej ma wysokość $\Theta(n)$?

4. (Zad. 09, 06.2017) Rozważamy B drzewa, których wierzchołki mogą pamiętać od dwóch do czterech kluczy. Narysuj, jak będzie wyglądać takie B drzewo po wstawieniu do początkowego pustego drzewa kolejno klucz $1, 2, \ldots, 10$.

Drzewa Splay

1. (Zad. 3, cz. 1, 06.2017) Narysuj

   - drzewo *Splay* po wykonaniu na początkowo pustym drzewie ciągu operacji:

$$insert(n), insert(n-1), insert(n-2), ..., (insert(1),$$

- drzewo *Splay* po wykonaniu operacji *Splay(n), Splay(n-1)* na drzewie otrzymanym w poprzednim punkcie

2. (Zad. 6, cz. 1, 06.2016) Czy trójelementowe drzewo złożone z korzenia i dwóch jego synów może być drzewem splay? Odpowiedź uzasadnij.

Haszowanie

1. (Zad. 6, cz. 1, 06.2017) Rozważamy haszowanie metodą adresowania otwartego, w której konflikty rozwiązujemy metodą liniową. Pokaż, że po umieszczeniu $n/2$ kluczy w tablicy n elementowej, mogą istnieć dwie lokalizacje w tej tablicy, do których kolejny (tj. $(n/2+1)$szy) klucz ma szansę trafić z prawdopodobieństwem $1/n$.

2. (Zad. 15, cz. 1, 06.2017) Ile pamięci zajmuje słownik statyczny (oparty o haszowanie dwupoziomowe) zawierający n kluczy? Co musimy w nim pamięta ć oprócz samych kluczy?

3. (Zad. 16, cz. 1, 06.2017) Podaj definicję i przykład uniwersalnej rodziny funkcji haszujących.

4. (Zad. 20, cz. 1, 06.2016) Jaka jest oczekiwana liczba kolizji podczas wstawiania n kluczy do tablicy o $m = n^2$ elementach, jeśli do wyznaczania miejsc wstawiania użyjemy funkcji o postaci $h(k) = ((ak + b) \bmod p) \bmod m$, gdzie:

5. (Zad. 12, cz. 1, 06.2015) Oszacuj prawdopodobieństwo, że nie będzie żadnej kolizji podczas haszowania funkcją z uniwersalnej rodziny $\sqrt{n}$ kluczy w tablicy rozmiaru n.

FFT

1. (Zad. 17, cz. 1, 06.2017) Algorytm FFT używaliśmy do zamiany reprezentacji wielomianu w reprezentację jako zbiór wartości wielomianu. Uzasadnij, dlaczego FFT możemy także zastosować do zamiany odwrotnej.

2. (Zad. 16, cz. 1, 06.2016) Jak wiadomo FFT jest algorytmem opartym na strategii Dziel i Zwyciężaj. Przedstaw redukcję wykonaną w tym algorytmie.

3. (Zad. 5, cz. 1, 06.2015) Przedstaw macierze dla transformacji Fouriera (???)

Algorytmy wyszukiwania wzorca: KMP, KMR, Shift-And

1. (Zad. 8, cz. 1, 06.2017) Czy istnieje wzorzec o długości n (dla dowolnego $n > 5$) nad alfabetem $\{a, b\}$, dla którego maksymalna wartość funkcji prefiksowej π jest równa

a 0,

b 1?

2. (Zad. 11, cz. 1, 06.2017) W algorytmie *Shift And* wykorzystywane są operacje logiczne na słowach maszynowych. Wytłumacz, w jaki sposób?

3. (Zad. 8, cz. 1, 06.2016) Dlaczego algorytm *Shift-And* stosowany jest jedynie do wyszukiwania krótkich wzorców?

4. (Zad. 10, cz. 1, 06.2016) Jaka jest największa wartość funkcji π dla wzorca $P = (ab)^k$? Odpowiedź uzasadnij.

5. (Zad. 1, cz. 1, 06.2015) Podaj przykład tekstu i wzorca dla których tablica C[0] = C[1] = C[9] = prawda, a dla pozostałych fałsz. Algorytm Shift-And.

6. (Zad. 2, cz. 1, 06.2015) Jak w KMR numeruje się słowa o długości 16?

7. (Zad. 4, cz. 1, 06.2015) Uzasadnij, że obliczenie funkcji pi(Wzorzec[1..m]) w algorytmie KMP ma złożoność O(m).

Algorytmy klasy NP i NC

1. (Zad. 5, cz. 1, 06.2016) Opisz ideę algorytmu klasy NC dla problemu dodawania liczb naturalnych.

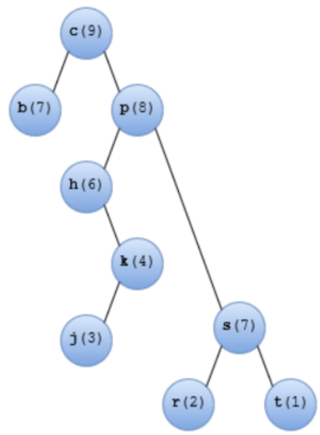
2. (Zad. 15, cz. 1, 06.2016) Podaj definicję problemu plecakowego z powtórzeniami i przedstaw pseudowielomianowy algorytm rozwiązujący ten problem. Uzasadnij, że jest on pseudowielomianowy.

3. (Zad. 13, cz. 1, 06.2015) Podaj pseudowielomianowy algorytm, który wypisuje dzielniki pierwsze liczby n.

Drzewce

1. (Zad. 4, cz. 1, 06.2017) Podaj przykład drzewca (tj. podaj wartość kluczy wraz z przydzielonymi im priorytetami) o n wierzchołkach, w którym każdy wierzchołek wewnętrzny ma tylko prawego syna. Następnie podaj, który wierzchołek będzie wymagał najwięcej rotacji podczas ustawiania go. Ile to będzie rotacji? Odpowiedź uzasadnij.

2. (Zad. 4, cz. 1, 06.2016) Narysuj ciąg rotacji, które zostaną wykonane w trakcie wykonywania **delete(p)** na poniższym drzewcu. Litery w wierzchołkach drzewca oznaczają klucze, a liczby w nawiasach - priorytety. Rotacje wypisz w kolejności wykonywania.



Rysunek 1: Drzewiec dla zadania 4.

Rysunek 1: rys do zad 4

3. (Zad. 8, cz. 1, 06.2015) Przedstaw drzewiec o n wierzchołkach, w którym usunięcie korzenia wymaga $\Omega(\sqrt{n})$ operacji, ew. podaj uzasadnienie dlaczego nie ma takiego drzewca.

Drzewa decyzyjne, gra z adwersarzem

1. (Zad. 5, cz. 1, 06.2017) Dolną granicę $\lceil \frac{3}{2}n - 2 \rceil$ na liczbę porównań niezbędnych do wyznaczenia *max* i *min* w zbiorze n elementów można wykazać stosując grę z adwersarzem. Opisz skuteczną strategię w takiej grze. Jeśli jest to strategia opisana na wykładzie, możesz na tym poprzestać. Jeśli jest to inna strategia, wykaż, że jest skuteczna.

Algorytmy wyboru k-tego elementu (mediany)

1. (Zad. 7, cz. 1, 06.2017) Opisz w jaki sposób wybierany jest *pivot* w każdym z następujących z następujących algorytmów znajdowania k-tego elementu:
   - Algorytm Hoare'a,
   - Algorytm Magicznych Piątek,
   - Lazy Select.
2. (Zad. 7, cz. 1, 06.2016) Opisz ideę algorytmu znajdowania mediany opartego na idei próbkowania losowego.
3. (Zad. 16, cz. 1, 06.2015) Przerób kod QuickSorta na QuickSelect (selekcja k-tego elementu zamiast sortowania). Jaką ma złożoność?

4. (Zad. 19, cz. 1, 06.2015) Podaj wzór rekurencyjny algorytmu magicznych piątek dla podziału na 7 elementów.

Izomorfizm drzew

1. (Zad. 10, cz. 1, 06.2017) Porównaj trudność problemu sprowadzania izomorfizmu drzew ukorzenionych i problemu sprawdzaniaa izomorfizmu drzew nieukorzenionych.

2. (Zad. 13, cz. 1, 06.2016) Przedstaw ideę szybkiego algorytmu sprawdzania izomorfizmu drzew. W jakim czasie działa ten algorytm?

Geometria obliczeniowa

1. (Zad. 14, cz. 1, 06.2017)W jaki sposób, stosując iloczyn wektorowy można sprawdzić, czy dwa punkty (powiedzmy p_1 i p_2) leżą po tej samej stronie prostej przechodzącej przez dwa punkty (powiedzmy A i B)?

Union Find

1. (Zad. 18, cz. 1, 06.2017) W analizie problemu *Union Find* wykorzystywaliśmy pojęcie rzędu wierzchołka oraz grupy rzędu. Przypomnij definicję tych pojęć. Ile maksymalnie bitów potrzebujemy przeznaczyć na pamiętanie rzędu w każdym wierzchołku?

2. (Zad. 14, cz. 1, 06.2016) W jakim czasie można wykonać ciąg n operacji **union** i **find**, w którym wszystkie operacje **union** poprzedzają operację **find**? Odpowiedź uzasadnij.

3. (Zad. 6, cz. 1, 06.2015) Podaj definicje: rząd wierzchołka i grupa rzędu wierzchołka

Drzewa Van Emde Boasa

1. (Zad. 19, cz. 1, 06.2017) Wyjaśnij po co oamiętane są wartości *min* i *max* w ja zdeh strukturze rekurencyjnej w drzewach (kolejkach van Emde Boasa)

2. (Zad. 9, cz. 1, 06.2016) Opisz (albo zapisz w pseudokodzie), w jaki sposób wykonywana jest operacja wstawiania klucza w drzewie van Emde Boasa.

Kopce - zwykłe, dwumianowe, Fibonacciego

1. (Zad. 2, cz. 1, 06.2017) Ile maksymalnie operacji *join* wykona się podczas łączenia kopców dwumianowych (wersja eager), z których każdy zawiera nie więcej niż 500 elementów? Przypomnienie: operacja *join* łączy dwa drzewa dwumianowe tego samego rzędu.

2. (Zad. 13, cz. 1, 06.2017) Niech T_1 oznacza najmniejsze pod względem liczby wierzchołków drzewo o rzędzie i, które może zawierać kopiec Fibonacciego. Narysuj drzewa T_i, dla $i = 0, 1, \ldots, 6$.

3. (Zad. 1, cz. 1, 06.2016) W jakim czasie można wykonać operację succ(x) w:

   - kopcu,
   - kopcu dwumianowym,
   - kopcu Fibonacciego,

   która znajduje następnik klucza znajdującego się w wierzchołku o adresie x? Przez następnik klucza k rozumiemy najmniejszy występujący w kopcu klucz k' taki, że k' taki, że $k' > k$. Jeśli$k$ jest największym kluczem w kopcu, to $k' = \infty$. Możesz założyć, że wszystkie klucze w kopc są unikalne. Odpowiedź uzasadnij.

4. (Zad. 17, cz. 1, 06.2016) Napisz w pseudokodzie szybką procedurę budowy kopca. W jakim czasie działa ta procedura?

5. (Zad. 18, cz. 1, 06.2016) Wyjaśnij, na czym polega operacja kaskadowego odcinania w kopcu Fibonacciego.

6. (Zad. 10, cz. 1, 06.2015) Ile jest maksymalnie drzew w kopcu:

7. (Zad. 11, cz. 1, 06.2015) Złożoność procedury budującej kopiec (wersja z przesun-do-gory()).

8. (Zad. 17, cz. 1, 06.2015) Porównanie kosztów operacji min, delmin, insert, meld dla kopców dwumianowych w wersji Lazy i Eager.

9. (Zad. 18, cz. 1, 06.2015)

   - Podaj definicje rzędu drzewa w kopcu Fibbonaciego,
   - Podaj górne ograniczenie na ten rząd,
   - Podaj ideę dowodu tego ograniczenia.

Algorytmy znajdowania MST

- (Zad. 11, cz. 1, 06.2016) W jakim czasie działa algorytm Kruskala, jeśli:

  - krawędzie podane są w kolejności rosnących wag„
  - kolejka priorytetowa zaimplementowana jest przy pomocy kopca Fibonacciego.

  Odpowiedź uzasadnij. *Uwaga: Oba te warunki są spełnione jednocześnie*

- (Zad. 20, cz. 1, 06.2015) Przykład grafu pełnego o n wierzchołkach takiego, że algorytm Boruvki znajdzie MST w jednej fazie.

Różne algorytmy

1. (Zad. 1, cz. 1, 06.2017) Opisz algorytm oparty na programowaniu dynamicznym wyznaczający optymalną kolejność mnożenia macierzy. Jaka jest jego złożoność? Jeśli jest to algorytm podany na wykładzie, możesz na tym poprzestać, w przeciwnym razie uzasadnij jego poprawność i złożoność.
2. (Zad. 20, cz. 1, 06.2017) Przypomnij sobie algorytm oparty na zasadzie *Dziel i zwyciężaj*, dla problemu znajdowania najbliższej pary punktów na płaszczyźnie. Opisz trzecią fazę algorytmu, a więc tę, która następuje po wywołaniach rekurencyjnych. Jaka jest jej złożoność?
3. (Zad. 12, cz. 1, 06.2016) Zapisz w pseudokodzie algorytm wielomianowy, znajdujący minimalny koszt obliczenia iloczynu ciągu macierzy.
4. (Zad. 15, cz. 1, 06.2015) Podaj pseudokod rozwiązania problemu LCS.

Inne

1. (Zad. 2, cz. 1, 06.2016) Rozwiąż rozwiązanie rekurencyjne (z redukcją do pierwiastka):
$$T(n) = \begin{cases} 1 & : \quad n = 1 \\ T(\sqrt{n} + O(1) & : \quad wpp \end{cases}$$
Możesz ograniczyć się do rozwiązania dla n mających odpowiednią postać (taką, by w trakcie redukcji argumenty dla T były liczbami naturalnymi).
2. (Zad. 3, cz. 1, 06.2016) Narysuj sieć Benesa-Waksmana dla n = 8.
3. (Zad. 3, cz. 1, 06.2015) Przedstaw graficznie sieć komparatorów o głębokości ≤ 4 sortującej wszystkie ciągi 0-1 o długości 7.
4. (Zad. 9, cz. 1, 06.2015) Podaj rekurencyjny wzór na T(n) tak, by jego rozwiązanie było $O(\log \log n)$.
5. (Zad. 7, cz. 1, 06.2015) Czy drzewa A i B będą miały równą wysokość, jeśli przeprowadzi się na nich n operacji insert o wartościach:
   - dla A: 1, 2, 3, ..., n
   - dla B: n, n-1, ..., 2, 1

Zadania z części 2.

1. (Zad. 1, cz. 2, 06.2017)
2. (Zad. 2, cz. 2, 06.2017)
3. (Zad. 3, cz. 2, 06.2017)

Zadania z części 3.

1. (Zad. 1, cz. 3, 06.2017) Ciąg nazywamy nienudnym, jeżeli każdy jego spójny podciąg zawiera co najmniej jedne unikalny element (tzn. występujący w tym podciągu dokładnie raz). Ułóż algorytm, który dla danego ciągu liczb naturalnych sprawdzi, czy jest on nienudny.

2. (Zad. 2, cz. 3, 06.2017) Wariancją ciągu liczbowego $A = \langle a_1, ..., a_n \rangle$ nazywamy liczbę

$$V_A = \begin{cases} 0 & gdy \quad n = 1 \\ \sum_{i=1}^{n-1} |ai + 1 - a_i| & gdy \quad n > 1 \end{cases}$$

ułóż algorytm, który dla zadanego ciągu liczbowego A znajdzie podział zbioru indeksów $\{1, 2 \ldots, n\}$ na dwa rozłączne podzbiory $I = \{i_1, i_2 \ldots, i_k\}$ oraz $J = \{j_1, j_2 \ldots, j_{n-k}\}$ takie, że:

- $i_1 < i_2 < \ldots < i_l$, $j_1, j_2, \ldots, j_{n-k}$ oraz $I \cup J = \{1, 2 \ldots, n\}$
- suma $V_{A_I} oraz V_{A_J}$ jest minimalna,

gdzie $A_I = \langle x_{i_1}, \ldots, x_{i_k} \rangle$, a $A_J = \langle x_{j_1}, \ldots, x_{k_{n-k}} \rangle$,

3. (Zad. 3, cz. 3, 06.2017) Rozważamy ciąg operacji *Insert(i)*, *DeleteMin* oraz *Min(i)* wykonywanych na S-podzbiorze zbioru $\{1, \ldots, n\}$. Obliczenia rozpoczynami z $S = \phi$. Instrukcja *Insert(i)* wstawia liczbę i do S. Instruckja *DeleteMin* wyznacza najmniejszy element w S i usuwa go z S. Natomiast wykonanie *Min(i)* polega na usunięciu z S wszystkich liczb mniejszych od i.
Niech ρ będzie ciągiem instrukcji $Insert(i)$, $DeleteMin$ oraz $Min(i)$ takimi, że dla każdego i, $1 \leq i \leq n$, instrukcja $Inser(i)$ występuje co najwyżej jeden raz. Mając dany ciąg ρ naszym zadaniem jest znaleźć ciąg liczb usuwanych kolejno przez instrukcję $DeleteMin$. Podaj algorytm rozwiązujący to zadanie.
Uwaga: *Zakładamy, że cały ciąg ρ jest znany na początku, czyli interesuje nas wykonanie go off-line*

4. (Zad. 4, cz. 3, 06.2017) Rozpiętością ciągu liczbowego $A = \langle a_1, ..., a_n \rangle$ nazywamy liczbę:

$$Span(A) = max\{a_i | i = 1, \ldots, n\} - min\{a_i | i = 1, \ldots, n\}$$

Ułoż algorytm obliczający

$$\sum_{1 \leq p \leq k \leq n} Span(A_p^k)$$

gdzie $A_p^k = \langle a_p, \ldots, a_k \rangle$