Egzamin AiSD 2015, część 1

1. **Opisz ideę algorytmu znajdowania mediany opartego na idei próbkowania losowego**

14.2.2 Algorytm *LazySelect*

IDEA Ze zbioru S wybieramy losowo próbkę R. Próbka powinna być niezbyt liczna, by można było szybko ją posortować. Znajdujemy w R dwa elementy L i H, takie że z dużym prawdopodobieństw em podzbiór $P \subseteq S$, elementów większych od L i mniejszych od H, jest nieduży oraz szukany element należy do P (możemy go wówczas łatwo znale"xć po posortowaniu P).

2. **Jaki jest czas działania poniżej zdefiniowanej funkcji MemoryTest…**
   [zadanie skreślone]

3. **Dlaczego algorytm Shift-And stosowany jest jedynie do wyszukiwania krótkich wzorców?**

Algorytm pamięta stan jako liczbę binarną. Aby wykonywanie operacji && || i << było szybkie i możliwe wzorzec musi być krótki i imieścić się w jednym słowie maszynowym.

4. **Podaj lemat zerojedynkowy oraz ideę jego dowodu.**

Lemat zerojedynkowy: jeśli sieć sortująca działa poprawnie dla wszystkich ciągów składających się tylko z zer i jedynek to poprawnie sortuje wszystkie ciągi dowolnych wartości.

Dowód nie wprost, korzystający z faktu, że jeśli dla ciągu <a1, a2, …, an> sieć sortująca wyznacza ciąg wyjściowy <b1, b2, …, bn>, to dla ciągu <f(a1), f(a2), …, f(an)> wyznaczy <f(b1), f(b2), …, f(bn)> (dla dowolnej funkcji f niemalejącej).

5. **Jakiej struktury danych użyjesz do implementowania kolejki priorytetowej, która będzie pamiętała klucze ze zbioru [1,2...n] oraz umożliwiać wykonanie operacji kolejkowych w czasie O(loglogn) ? Opisz ideę jej konstrukcji.**

6. **Rozważamy słownik statyczny realizowany przy pomocy haszowania dwupoziomowego. Załóżmy, żę dla pewnych danych pierwotna funkcja haszująca umieszcza w k-tym kubełku u elementów. Jaki jest rozmiar tablicy (wtórnej), w której będą te elementy umieszczane? Wyjaśnij dlaczego ten rozmiar jest taki a nie inny?**

7. **Przedstaw ideę szybkiego algorytmu sprawdzania izomorfizmu drzew. W jakim czasie działą ten algorytm?**

PROBLEM:
Dane: T_1, T_2 -drzewa o ustalonych korzeniach,
Zadanie: sprawdzić czy T_1 i T_2 są izomorficzne.

IDEA: Wędrując przez wszystkie poziomy (począwszy od najniższego) sprawdzamy czy na każdym poziomie obydwa drzewa zawierają taką samą liczbę wierzchołków tego samego typu. (Wierzchołki są tego samego typu, jeśli poddrzewa w nich zakorzenione są izomorficzne.)

Czas O(n

8. **W jakim czasie można wykonać ciąg n operacji Union-Find w którym wszystkie operacje union poprzedzają operacje find. Odpowiedź uzasadnij.**
   ODP: Załóżmy, że operacje union nie wywołują funkcji find(nie wykonują kompresji). Najgłębsze drzewo uzyskamy jeśli cały czas będziemy łączyć drzewa o tej samej randze(wysokości). Najpierw będziemy zbiory/drzewa wielkości 1, potem 2…. Maksymalna głębokość takiego drzewa to lg(n). Teraz przejdzemy do analizy zamortyzowanej. Koszt UNION: 2, FIND:1. Robiąc union przegranemu(temu który zostanie poniżej) wierzchołkowi przypisujemy 1 jednostkę(a drugą spalamy na przepięcie wskaźników itp). Biorąc najgorsze drzewo, mamy sytuację kiedy wszystkie wierzchołki poza korzeniem mają 1 jedostkę na sobie. Robiąc find wykorzystujemy 1 jednostkę na przeskoczenie wyżej (i późniejsze przepięcie wskaźnika). Jeśli jesteśmy w czymś niżej niż 2 użyjemy jednostek z wierzchołków. Będąc w korzeniu lub w jego pierwszych synach wystarczy nam jednostka z operacji find.
   Koszt wszystkich operacji to 2*(|U|) + Ilość_FIND = **O(n).**

9. **Dla danego zestawu n kluczy a1,a2...an i przydzielonych im priorytetów p1,p2...pn tworzymy drzewiec. Dla których z poniższych przypadków drzewiec ten będzie wyznaczony jednoznacznie?**
   a. **Wszystkie klucze są różne, ale istnieje jedna para równych priorytetów**
   b. **Wszystkie priorytety są różne, ale istnieje jedna para równych kluczy**
   c. **Wszystkie klucze są różne i wszystkie priorytety są różne**

   C dla a) i b) łatwo znaleźć kontryprzykłady.
10. **Jak można pogodzić istnienie dolnej granicy Ω(nlogn) na sortowanie ciągu n liczb z tym, że sortowanie przez zliczanie wykonuje mniej niż nlogn operacji?**
    Ta dolna granica dotyczy algorytmów które wykonuje tylko porównania na elementach.

11. **Podaj definicję i przykład uniwersalnej funkcji haszującej**

Rodzina funkcji hashujacych H jest uniwersalna, jeśli dla dwoch dowolnych kluczy k,l nalezacych do U, liczba funkcji hashujacy h z H takich, ze h(k) = h(l) wynosi co najwyzej |H|/m.

Czyli jesli losowo wybierzemy h z tej rodziny, to szansa na kolizje pary kluczy wynosi 1/m.

m - zbior wartosci funkcji hashujacej

p - liczba pierwsza, p > m

H_ab(k) = ((ak + b) mod p) mod m), gdzie a,b miedzy 1 .. p-1

12. **Podaj definicję problemu plecakowego z powtórzeniami i przedstaw pseudowielomianowy algorytm rozwiązujący ten problem**

http://www.geeksforgeeks.org/dynamic-programming-set-10-0-1-knapsack-problem/

13. **FFT jest algorytmem opartym na strategii dziel i zwyciężaj. Przedstaw redukcję wykonywaną w tym algorytmie**

- Idea algorytmu.

  Niech
  $A^{[0]}(z) = a_0 + a_2 z + a_4 z^2 \ldots + a_{n-2} z^{n/2-1}$ i $A^{[1]}(z) = a_1 + a_3 z + a_5 z^2 \ldots + a_{n-1} z^{n/2-1}$.
  Wówczas $A(x) = A^{[0]}(x^2) + xA^{[0]}(x^2)$.

  Tak więc problem obliczenia wartości wielomianu A stopnia $n-1$ w n punktach: $\omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1}$, redukuje się do problemu obliczenia wartości dwóch wielomianów $A^{[0]}$ i $A^{[1]}$ stopnia $\frac{n}{2}-1$ w $\frac{n}{2}$ punktach: $\omega_{n/2}^0, \omega_{n/2}^1, \ldots, \omega_{n/2}^{(n/2)-1}$.

14. **Niech H będzie kopcem dwumianowym (wersja eager) zawierającym 1000000 elementów. Ile najmniej elementów może zawierać kopiec dwumianowy G, tak by po połączeniu G z H wynikowy kopiec zawierał**
    a. **Jedno drzewo dwumianowe**
    b. **Dwa drzewa dwumianowe**
    c. **Dziewiętnaście drzew dwumianowych?**

15. **Napisz w pseudokodzie szybką procedurę budowy kopca. W jakim czasie działa ta procedura?**
    Kopiec można zbudować naiwnie w nlog(n): (1..n) do insert_node(x) end
    Optymalniej jest miec gotowa tablice A z elementami(jakby 1 elementowe kopce) i przerobic to na jeden kopiec takim kodem (n/2.. 0) do |i| w_dol(A, i) end w czasie n. (Jest dobrze opisane w notatkach Lorysia)

16. **Załóżmy, że przełączniki w sieci przełączników ustawiane są losowo (z rozkładem jednostajnym). Dla jakich n naturalnych istnieje sieć przełączników o n wejściach, która z jednakowym prawdopodobieństwem wygeneruje każdą permutację n-elementową? Uzasadnij.**

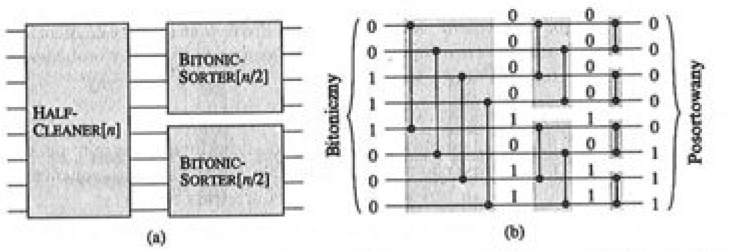
17. **Na czym polega operacja kaskadowego odcinania w kopcach Fibonacciego?**

18. **Podaj przykład danych, dla których Algorytm Dijkstry poda niepoprawny wynik.**

19. **Który z poniższych algorytmów sortowania, może w najgorszym przypadku wykonać Omega(n^2) porównań:**
    a. **Quicksort**
    b. **Mergesort**

c. **Insertsort**
d. **Selectsort**
e. **Heapsort**
Quicksort jeśli słabo wybieramy mediane.InsertSort i SelectSort robią to z definicji

20. **Przedstaw, w jaki sposób sieci półczyszczące są wykorzystywane w konstrukcji bitonicznych sieci sortujących.**



Rys. 28.9. Sieć porównująca BITONIC-SORTER[n] zilustrowana tutaj dla $n = 8$. (a) Konstrukcja rekurencyjna: sieć HALF-CLEANER [n] połączona z dwoma równoległymi kopiami sieci BITONIC-SORTER [$n/2$]. (b) Sieć po rozwinięciu rekurencji. Każda składowa HALF-CLEANER została zacieniowana

Egzamin AiSD 2014, część 1

1. **Podaj przykład wzorca i tekstu, dla których tablica intR[10] wyliczana w algorytmie Shift-And może przyjąć następujące wartości**
   R[0] = R[1] = R[9] = 1
   Oraz R[i] = 0 dla pozostałych i, lub uzasadnij że taki wzorzec i tekst nie istnieją.

2. **Jednym z elementów algorytmu KMR jest numerowanie podsłów danego słowa. Przedstaw, w jaki sposób dokonywana jest numeracja podsłów długości 16.**

3. **Przedstaw graficznie sieć komparatorów o głębokości nie większej niż 4, która sortuje wszystkie ciągi długości 7, złożone z zer i jedynek, lub uzasadnij, dlaczego taka sieć nie istnieje.**

4. **Poniżej przedstawiony jest algorytm obliczający funkcję prefiksową π dla algorytmu KMP. W jakim czasie działa ten algorytm? Uzasadnij.**

```
Procedure Compute-Prefix-Function(P)
        m <- length(P)
        π(1) <- 0
        k <- 0
```

```
for q<-2 to m do
      while k>0 and P[k+1]≠P[q] do k <- π(k)
      if P[k+1]=P[q] then k <- k+1
            π(q) <- k
Return π
```

5. Transformację Fouriera można przedstawić jako przekształcenie liniowe wektora. Przedstaw macierz tego przekształcenia oraz macierz przekształcenia odwrotnego.

6. Podaj definicję pojęć: rząd wierzchołka, grupa rzędu.

7. Rozważ wstawienie sekwencji kluczy do początkowo pustego drzewa splay:
   a. 1,2,3...n-1,n
   b. N,n-1...3,2,1

   W którym przypadku powstanie wyższe drzewo?

8. Przedstaw drzewiec o n wierzchołkach, w którym usunięcie korzenia wymaga wykonania Omega(pierwiastek z n) rotacji, lub uzasadnij dlaczego taki drzewiec nie istnieje.

9. Przymierzasz się do rozwiązania pewnego problemu i celujesz w algorytm działający w czasie Theta(loglogn). Rozważasz zastosowanie metody dziel i zwyciężaj, więc złożoność twojego algorytmu będzie się wyrażać zależnością rekurencyjną

   T(n)= O(1) dla n<const
   T(n)=aT(f(n))+g(n) w p.p.

   Podaj jakie wartości może przyjąć stała a, oraz jakie mogą być funkcje f i g?

10. Ile maksymalnie drzew może znaleźć się w:
    a. Kopcu dwumianowym(wersja eager)
    b. Kopcu fibonacciego

    Zawierającym n elementów

11. Jaka jest złożoność poniższej funkcji tworzenia kopca w tablicy K:

```
procedure buduj-kopiec(K[1...n])
      for i<-2 to n: przesun-wyzej(K,i)
```

gdzie przesun-wyzej przywraca porządek kopcowy w kopcu zapamiętanym w tablicy K[1...i], o ile jest on zaburzony przez element K[i].

12. **Podaj sensowne oszacowanie prawdopodobieństwa tego, że nie wystąpią żadne kolizje podczas umieszczania pierwiastek z n kluczy w tablicy n-elementowej przy pomocy funkcji haszującej n, losowo wybranej z rodziny uniwersalnej.**

13. **Podaj pseudowielomianowy algorytm, znajdujący rozkład liczby naturalnej na czynniki pierwsze. Uzasadnij stwierdzenie, że jest pseudowielomianowy.**

14. **Rozważ modyfikację drzew AVL, w której wysokość lewego i prawego poddrzewa zakorzenionych w tym samym wierzchołku mogą się różnić o nie więcej niż 2. Udowodnij, że każde takie drzewo z n wierzchołkami ma wysokość Theta(log n)**

15. **Zapisz w pseudokodzie algorytm znajdujący długość najdłuższego wspólnego podciągu dwóch ciągów.**

16. **Zmień poniżej zapisany pseudokod algorytmu QuickSort tak, aby rozwiązywał problem znajdowania k-tego co do wielkości elementu tablicy A. Jaka jest złożoność twojego algorytmu?**

17. **Porównaj koszt wykonywania operacji min, deletemin, insert, meld w wersji lazy i w wersji eager kopców dwumianowych.**

18. **Podaj definicję rzędu drzew w kopcach Fibonacciego**
    a. **Górne ograniczenie na ten rząd**
    b. **Ideę dowodu tego ograniczenia**

19. **Podaj wzór rekurencyjny na liczbę porównań "magicznych piątek" w którym ciąg wejściowy dzielimy na 7 zamiast 5.**

20. **Podaj przykład grafu pełnego o n wierzchołkach, dla którego Borovka znajduje MST w jednej fazie.**