

Część I

I egzamin cząstkowy 2014

Rozdział 1

zagadnienia

- Algorytmy
 - mnożenie po rosyjsku
 - szybkie potęgowanie
 - szybkie obliczanie liczb Fibonacciego
 - algorytmy sortowania: select i insert
 - maszyna RAM jako model obliczeń
 - notacja asymptotyczna
- kopiec
 - definicja
 - implementacja tablicowa
 - realizacja operacji kopcowych
 - zastosowania: heapsort, kolejka priorytetowa
 - kopiec minmax
- Algorytmy zachłanne
 - problem wydawania reszty
 - algorytmy znajdowania minimalnego drzewa rozpinającego
 - * algorytm Prima
 - * algorytm Kruskala
 - * algorytm Boruvki
 - szeregowanie zadań
 - szeregowanie zadań z deadlineami
- Metoda Dziel i Zwyciężaj
 - algorytmy sortowania (mergesort, quick sort)
 - algorytmy mnożenia długich liczb (Karatsuby)
- Metoda Dziel i Zwyciężaj (cd)
 - algorytmy mnożenia długich liczb oparte na podziale na wiele części
 - jednoczesne znajdowanie maksimum i minimum
 - sieć permutacyjna Waksmana-Benesa
 - znajdowanie najbliższej pary punktów na płaszczyźnie
 - zachłanny algorytm aproksymacyjny dla problemu Set Cover

Rozdział 2

potencjalne zadania

2.1.

Przedstaw ideę algorytmu Boruvki (Sollina).

G - nasz graf G' - nasza odpowiedź, początkowo tylko wierzchołki z G i żadnych krawędzi Wykonujemy kroki:

1. Dla każdego wierzchołka w G znajdź najkrótszą incydentną krawędź i dodaj ją do G'
2. Utwórz nowy graf G' , w którym wierzchołkami są spójne składowe w starym G'

Iterujemy do póki nie zostanie jeden wierzchołek. Wszystkie dodane krawędzie do G' utworzą odpowiedź.

2.2.

Które z poniższych algorytmów mogą działać niepoprawnie dla grafów z ujemnymi wagami krawędzi? Odpowiedź uzasadnij.

- a) algorytm Kruskala
- b) algorytm Prima
- c) algorytm Dijkstry

a) Kruskala - działa dobrze

b) Prima - działa dobrze

c) Dijkstra - Jeżeli gdzieś w grafie występuje cykl o negatywnej wadze, to wszystkie drogi nie mają najkrótszej ścieżki. Algorytm może tego nie wykryć, bo nigdy nie wraca do wierzchołków już rozważonych, a cykl możemy znaleźć dopiero pod koniec wykonania, dlatego zwróci jakieś wartości dla wierzchołków, a nie powinien.

2.3.

Rozwiąż równanie rekurencyjne

$$T(1) = 1$$

$$T(2) = 3$$

$$T(n) = T(n-2) + 2n - 1 \quad \text{dla } n > 2$$

$$T(1) = 1$$

$$T(2) = 3$$

$$T(n) = T(n-2) + 2n - 1 = T(n-2) + n + (n-1) \quad \text{dla } n > 2$$

$$\begin{aligned} T(n) &= n + (n-1) + T(n-2) = \\ &= n + (n-1) + (n-2) + (n-3) + T(n-4) = \\ &= n + (n-1) + (n-2) + (n-3) + \dots + 4 + 3 + 2 + 1 = \\ &= \frac{n(n+1)}{2} \end{aligned}$$

2.4.

Który z poniższych algorytmów sortowania może w najgorszym przypadku wykonać $\Omega(n^2)$ porównań:

- a) quicksort
- b) mergesort
- c) insertsort

Przypomnienie: $\Omega(n^2)$ oznacza nie mniej niż cn^2 dla pewnej stałej $c > 0$.

- a) quicksort - jeżeli znajdowanie pivotu jest źle zrobione, tzn. odcina stałą liczbę elementów przy każdym partition, np. stosunek 1 : (n-1), to wtedy złożoność to $n * O(n) = O(n^2)$
- b) mergesort - worst case $O(n \log n)$
- c) insertsort - posortowany lub odwrotnie posortowany ciąg będzie mieć $O(n^2)$

2.5.

Napisz procedurę partition (nie musi być to wersja z wykładu, ale musi być efektywna).

Pivot jest w A[p], funkcja zwraca granicę podziału

```

Part(A[1..n], p, r)
  x <- A[p]
  i <- p-1
  j <- r+1

  while(i<j) do
    repeat(j--) until A[j] <= x
    repeat(i++) until A[i] >= x

    if(i<j) swap(A[i], A[j])
    else return j

```

2.6.

Przedstaw strategię zachłanną algorytmu aproksymacyjnego dla problemu Set Cover o współczynniku aproksymacji H_n .

Problem: Wybrać najmniejszy taki podzbiór z rodziny zbiorów, aby wszystkie pojedyncze elementy z całej rodziny znalazły się w tym podzbiorze. Algorytm aproksymacyjny: W każdym kroku odrzucamy zbiór, który nie pokrywa jak największej części elementów.

U – uniwersum do pokrycia

R – wejściowa rodzina zbiorów

W – wynikowa rodzina zbiorów na początku pusta

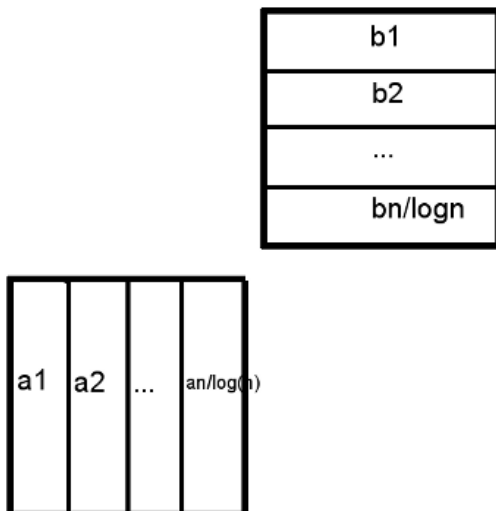
```

while U jest niepusty
  wybierz takie Z z R ze Z przekroj U jest maksymalne
  dodaj Z do rodziny wynikowej W
  U \= Z
  usun Z z R

```

2.7.

W algorytmie czterech Rosjan obliczane są iloczyny macierzy o rozmiarze $n \times \log n$ i macierzy o rozmiarze $\log n \times n$. Ile takich iloczynów jest obliczanych?



Liczymy iloczyny macierzy dla każdego $a_i \cdot b_i$. Takich iloczynów jest oczywiście $\frac{n}{\log(n)}$, a każdy w wyniku daje jedną macierz kwadratową. Jeśli zsumujemy te macierze to otrzymamy iloczyn, o który nam chodzi.

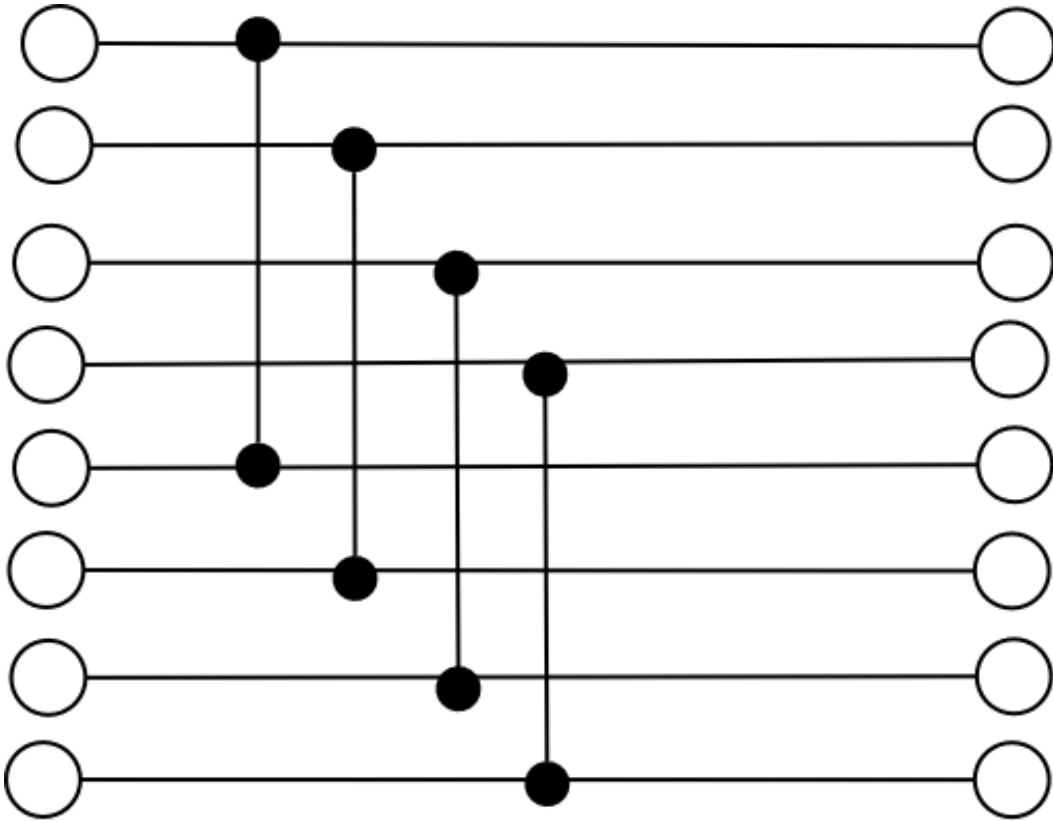
2.8.

Opisz, w jaki sposób DFS może być zastosowane do znalezienia cyklu Eulera w grafie.

Graf przechodzimy przy pomocy rekurencyjnej procedury DFS. Przebyte krawędzie usuwamy, a wierzchołki po zakończeniu przetwarzania umieszczamy na początku kolejki. Jeśli graf posiada cykl Eulera, to po zakończeniu algorytmu w kolejce znajdą się kolejne wierzchołki tego cyklu.

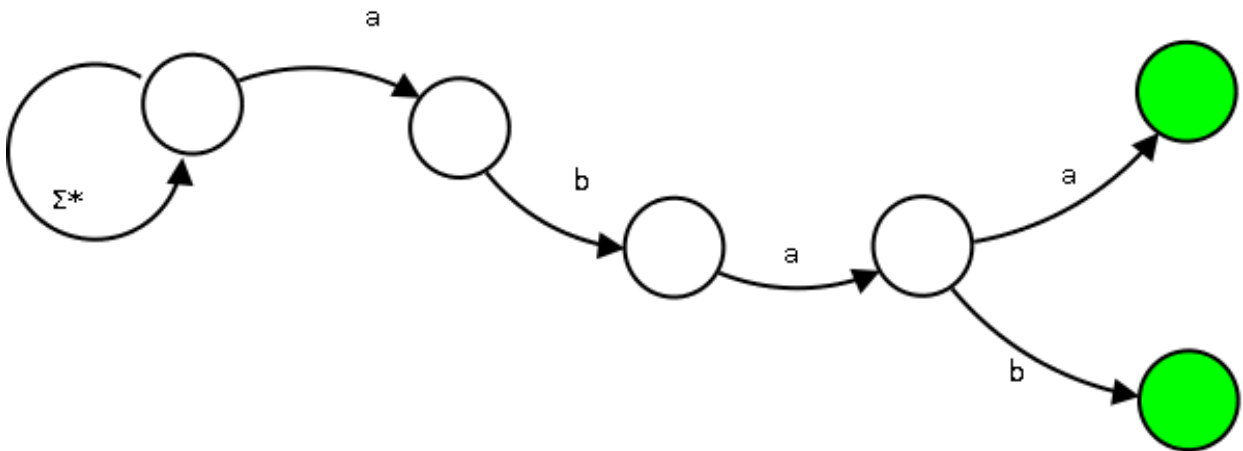
2.9. ??? tego chyba nie było jeszcze

Narysuj sieć połącyszczącą o ośmiu wejściach.



2.10. ??? tego chyba nie było jeszcze

Narysuj automat rozpoznający *abaa* i *abab*.



2.11.

W jaki sposób problem mnożenia macierzy może być wykorzystany do rozwiązania problemu najkrótszych ścieżek w grafie?

Mamy grafy A i B w postaci macierzy adiacencji.

Możemy korzystać z algebry, w której:

'+' to działanie min

'.' to dodawanie

Wtedy jedno mnożenie macierzy, to jeden krok relaksacji najkrótszych ścieżek, musimy wykonać n takich relaksacji. Złożoność $O(n^4)$. Można poprawić złożoność. Nie da się zastosować algorytmu Strassena, bo nie mamy działania odwrotnego do min. Można za to skorzystać z szybkiego potęgowania, co zmniejsza nam koszt do $O(n^3 \log n)$.

2.12.

Opisz, w jaki sposób obliczenie wielomianu n -tego stopnia w n -tych pierwiastkach jedności jest redukowane do obliczenia dwóch wielomianów stopnia $n/2$ w $(n/2)$ -tych pierwiastkach jedności.

Mamy wielomian $A(x)$ stopnia n , reprezentowany przez współczynniki a_0, a_1, \dots, a_n .

Tworzymy 2 nowe wielomiany:

$$A^{[0]}(x) = a_0 + a_2x + a_4x^2 + \dots a_{n-2}x^{n/2-1} + a_nx^{n/2}$$

$$A^{[1]}(x) = a_1 + a_3x + a_5x^2 + \dots a_{n-1}x^{n/2-1}$$

$$A(x) = A^{[0]}(x^2) + xA^{[1]}(x^2)$$

Rozdział 3

nierozwiązane

3.1.

Opisz algorytm mnożenia długich liczb (n bitowych)?

3.2.

Jak posortować n liczb z przedziału $[1, n^2]$ w czasie liniowym?

3.3.

Ile warstw musi mieć sieć przełączników, żeby dało się uzyskać wszystkie przesunięcia cykliczne ciągu n -elementowego?

Rozdział 4

zadania dodatkowe z list

4.1.

Co stałoby się z mocą obliczeniową maszyny RAM gdyby instrukcje ADD i MULT zostały usunięte repertuaru instrukcji? Jak zmieniłby się koszt obliczeń?

4.2.

Pokaż, że dla każdego programu maszyny RAM istnieje równoważny program maszyny RAM używający nie więcej niż 2^{14} komórek pamięci.

Bibliografia