Big Data
Lista zadań

Jacek Cichoń, WPPT PWr, 2016/17

1 Wstęp

Zadanie 1 — Znajdź źródła swojej ulubionej książki. Zapisz je w formacie utf-8. W tm zadaniu zastosuj swój ulubiony język programowania (plik ma być takich rozmiarów aby w całości mieścił się w pamięci komputera).

1. Wczytaj książkę i podziel je na słowa. Usuń z tej listy stop-words (możesz ja znaleźć na stronie `https://pl.wikipedia.org/wiki/Wikipedia:Stopwords`
2. Wyznacz częstotliwości występowania wszystkich słów. Masz zbudować listę postaci $[[slowo \Rightarrow liczba], [slowo \Rightarrow liczba], \ldots])$.
3. Posortuj otrzymaną listę według drugiego parametru.
4. Wyświetl kilkadziesiąt pierwszych elementów. Usuń z niej kilkanaście początkowych elementów i zapisz listę do pliku tekstowego.
5. Zbuduj chmurę wyrazów (word cloud) z otrzymanej listy. Możesz skorzystać np. z serwisu `http://www.wordclouds.com/`

Celem tego zadania jest wygenerowanie mniej więcej takiego obrazka (dla książki "Pan Tadeusz"):

Zadanie 2 — To jest kontynuacja poprzedniego zadania.

1. Podziel swoją książkę na rozdziały.
2. Każdy rozdział potraktuj jako dokumenty.
3. Podziel dokumenty na słowa. Wyznacz indeksy TF.IDF wszystkich słów we wszystkich rozdziałach
4. Zbuduj chmury wyrazów dla wszystkich rozdziałów i jedną chmurę dla całego dokument.

Zadanie 3 — Zainstaluj język Scala na swoim komputerze i pobaw się w konsoli REPL podstawowymi obiektami tego języka. Rozszyfruj i zapamiętaj skrót REPL

Zadanie 4 — Oprogramuj w języku Scala funkcje $\gcd$ (największy wspólny dzielnik) oraz lcm (najmniejsza wspólna wielokrotność). Ustalmy, że $\gcd(0,0) = \mathrm{lcm}(0,0) = 0$ oraz $\gcd(a,b) = \gcd(|a|,|b|)$ i $\mathrm{lcm}(a,b) = \mathrm{lcm}(|a|,|b|)$.

1. Oprogramuj następnie funkcję Eulera phi zdefiniowaną wzorem
$$\phi(n) = |\{k \in \{1,\ldots,n\} : \gcd(k,n) = 1\}|$$

Spróbuj zastosować (mocno nieefektywną w tym przypadku) metodę `count` do zakresu Range(1,n+1). Sprawdź, czy na pewno otrzymasz $\phi(1) = 1$.

2. Sprawdź poprawność napisanych funkcji obliczając
`Range(1,101).filter(100%_==0).map(x=>Euler(x)).sum`
w konsoli REPL. Powinno wyjść 100
3. Poznaj prosty dowód tego, że $\sum_{k|n} \phi(k) = n$ dla każdej liczby naturalnej $n \geq 1$.

Zadanie 5 — Zrealizuj Zadanie 1 w języku Scala.

1. Zaimportuj bibliotekę io.Source (`import scala.io.Source`)
2. Skorzystaj z polecenia `Source.fromFile(source, "UTF-8")` do wczytania pliku, zamień go na łańcuch (`mkString`) i następnie podziel na wyrazy (split("\\s+")). Możesz to zrobić jednym poleceniem.
3. Wczytaj stop-słowa i podziel je na wyrazy.
4. Usuń z książki stop słowa (coś w stylu `Book.filterNot(Stop.contains(_))`)
5. Pogrupuj słowa książki (coś w stylu `Filtered.groupBy(x=>x)`)
6. Zredukuj (coś w rodzaju `Groupped.mapValues(x=>x.length)`)
7. Posortuj według drugiego parametru (np. `reduced.toSeq.sortWith((x,y)=>x._2>y._2)`)
8. Zapisz wynik do pliku. Uwaga: możesz skorzystać z obiektu `PrintWriter` z bibliotek `java.io`

Zadanie 6 — Załóżmy, że mamy dostęp do bazy zakupów klientów w sieci hurtowni środków chemicznych z poprzedniego roku. W ciągu roku 10^7 klientów odwiedza ją 10 razy i za każdym razem kupuje średnio 10 różnego typu produktów z puli 200 dostępnych typów produktów. Załóżmy że znaleźliśmy w tej bazie danych dwóch klientów którzy zakupili choć raz ten sam koszyk produktów. Czy jest to czysty przypadek?

2 Funkcje haszujące

Zadanie 7 — Rozważmy funkcje haszującą zadaną wzorem $h(x) = x \mod 21$. Stosujemy ją do liczb podzielnych przez pewną stałą c. Dla jakich stałych c jest to odpowiednia funkcja haszująca, czyli dla jakich stałych c można się spodziewać, że rozkład załadowania kubełków $\{0, \ldots, 20\}$ będzie jednostajny?

Zadanie 8 — Znajdź wzór na rząd elementu $k \in \{0, \ldots, n-1\}$ w grupie $C_n = (\{0, \ldots, n-1\}, \oplus_n)$? Jaki jest związek tego zadnia z poprzednim zadaniem?

Zadanie 9 — Mamy n kubełków. Rzucamy do niech k kul.

1. Oszacuj k taki aby z dużym prawdopodobieństwem doszło do 3-kolizji, czyli aby a jakimś kubełku znalazły się 3 kulki.
2. Sprawdź eksperymentalnie otrzymany wynik
3. Uogólnij zadanie na a - kolizje

Zadanie 10 — Dwóch studentów ma dzban wypełniony 8 litrami napoju. Mają do dyspozycji dzbanek o pojemności 5 litrów oraz drugi dzbanek o pojemności 3 litrów. Chcą podzielić się równo napojem. Jak mogą to zrobić? Zagadnie to można potraktować jako system przepisujący o stanie początkowym $\{8, 0, 0\}$ Następujący kod (język Mathematica) opisuje pojedynczy, losowy krok transformacji stanu.

```
Move[C_]:= Block[{x,y,V={8,5,3},Kopia,suma},
  Kopia = C;
  {x,y} = RandomSample[{1,2,3},2]; (*Chcę przelać z x do y*)
  suma = C[[x]]+C[[y]];
  If[suma<=V[[y]],
    Kopia[[x]]=0;Kopia[[y]]=suma,
    Kopia[[x]]=suma-V[[y]];Kopia[[y]]=V[[y]];
  ];
  Kopia
]
```

Można go uruchomić w pętli, czekając aż osiągniemy stan $\{4, 4, 0\}$. Jednak jest to kiepskie rozwiązanie - algorytm taki wpada bardzo często w pętle. Zastosuj tablicę mieszającą (hashTable) do kontroli historii przebiegu tego

algorytmu (ma ona służyć do unikania zapętleń).
Wskazówka: możesz użyć np. java: java.util.Hashtable; Scala: scala.collection.mutable.Set.empty[List[Int]]; Python: np. set; Wszystkie te klasy są oparte na HashTables.

3 Model MapReduce

Na razie zadania programistyczne realizujemy w standardowych językach programowania (Jave,Python, Scala).

3.1 Działania

Zadanie 11 — Załóżmy, że $\star$ jest działaniem łącznym.

1. Pokaż, że $(a \star b) \star (c \star d) = a \star (b \star (c \star d)) = ((a \star b) \star c) \star d$.
2. Uogólnij to zadanie na dowolną liczbę zmiennych.
3. Ile różnych wyrażeń możesz zbudować dla pięciu zmiennych? Wskazówka: Może przydać się zapisanie tych wyrażeń w postaci drzew.

Zadanie 12 — Niech $x \oplus y = x + y + 1$ oraz $x \oplus y = xy + x + y$ dla $x, y \in \mathbb{R}$. Pokaż, że są to działania łączne i przemienne na $\mathbb{R}$. Wskazówka: Spróbuj to zrobić z minimalną liczbą rachunków.

Zadanie 13 — Podaj kilka przykładów działań nieprzemiennych. Podaj kilka przykładów działań które nie są łączne.

Zadanie 14 — Pokaż, że operacje $\min(x, y)$ i $\max(x, y)$ są przemienne i łączne. Czy operacja $s(x, y) = \frac{x+y}{2}$ jest łączna?

Zadanie 15 — Co robią następujące polecenia języka Python?

1. `list(filter(lambda x: x%2==0,range(1,100)))`
2. `list(map(lambda x: x*x,range(1,10)))`
3. `reduce(lambda x,y: x+y, [1,2,3,4,5])`
4. `reduce(lambda x,y: x*y, [1,2,3,4,5])`
5. `reduce(lambda x,y: x/y, [1,2,3,4,5])`

Uwaga: funkcję reduce zaimportuj z biblioteki **functions**.

3.2 Algorytmy MapReduce

Zadanie 16 — Wymień jakie aspekty działania systemu MapReduce są poza zasięgiem programisty. Które elementy kontroluje programista?

Zadanie 17 — Zaprojektuj algorytm MapReduce który dostaje bardzo duży zbiórliczb całkowitych i produkuje na wyjściu:

1. Największą liczbę.
2. Średnią wszystkich liczb.
3. Ten sam zbiór ale bez powtórzeń.
4. Liczbę różnych elementów bez powtórzeń.

Zadanie 18 — (Odwrócenie grafu) Dany jest graf w postaci listy sąsiadów: $[w, [w_i, w_{i1}, w_{i2}, \ldots, w_{i,n_i}]]$ zapisany w zbiorze tekstowym,np

```
[
  [1, [3,4,5]],
  [2, [1,3]],
  [3, [4,5]],
  [4, [1,2]],
  [5, [4,5]]
]
```

Zastosuj technologię MapReduce do zbudowania grafu z odwróconymi linkami.
Wskazówka: Jeśli programujesz w języku Python, to możesz skorzystać z funkcji `groupby` z biblioteki **itertools**; pamiętaj, że lista par którą chce się grupowac musi być posortowana. W języku Scala jest jeszcze łatwiej: przyjrzyj się metodzie groupBy stosowalnej do klasy Traversable.

Zadanie 19 — (Częste produkty) Mamy dany duży zbiór koszyków zakupowych z hipermarketu. Wyznacz zbiór wszystkich częstych par, czyli takich par produktów, które często występują występują w jednym koszyku. Załóżmy, że zbiór wszystkich możliwych par występujących w jednym koszyku jest tak duży, że nie jesteśmy w stanie ich wszystkich przetworzyć w realnym czasie.
Wskazówka: Jeśli para jest częsta to i każdy z jej składników jest częsty.

Zadanie 20 — (Odwrócony Indeks) Mając dany zbiór dokumentów zbuduj inverted index słów w nich występujących.

Zadanie 21 — Zaprojektuj algorytm MapReduce, który wyznacza złączenie dwóch relacji o schamacie R(A,B,C) i S(X,Y,Z) według połączenia B=X oraz C=Y, czyli wyznacz tabelę

$$\{(A,Y) : (\exists B,C)(R(A,B,C) \wedge S(B,C,Y)\} \ .$$

Zadanie 22 — W pliku `TwoCollisions.csv`, do którego link znajduje się na stronie wykładu, w każdej linijce znajduje się `(NumerHotelu, NumerDnia,NumerOsoby)`. Znajdź takie osoby, które w dwóch różych dniach znajdowały się się w tym samym hotelu.

Zadanie 23 — Niech $F : ((\mathbb{N}\setminus\{0\})\times\mathbb{R})^2 \to (\mathbb{N}\setminus\{0\})\times\mathbb{R}$ będzie funkcją określoną wzorem

$$F([c_1,x_1],[c_2,x_2]) = [c_1+c_2, \frac{c_1x_1+c_2x_2}{c_1+c_2}]$$

1. Pokaż, że F jest działaniem przemiennym i łącznym.
2. Oznaczmy przez $\oslash$ działanie $x \oslash y = F(x,y)$. Znajdź zwartą formułę dla

$$[c_1,x_1]\oslash[c_2,x_2]\oslash\ldots\oslash[c_n,x_n] \ .$$

3. Zastosuj tę własność funkcji do zastosowania combainera dla problemu wyznaczania średniej i wariancji.

Zadanie 24 — Zastosuj metodę map-reduce do wyznaczenia średniej geometrycznej i harmonicznej.

Zadanie 25 — Zastosuj metodę map-reduce do wyznaczenia wszystkich anagramów występujących w zbiorze tekstowym.

Zadanie 26 — Multizbiorem o skończonym nośniku Ω nazywamy funkcję $F:\Omega\to\mathbb{N}$. Dla $F,G:\Omega\to\mathbb{N}$ określamy $(F\cup G)(\omega)=\max\{F(\omega),G(\omega)\}$, $(F\cap G)(\omega)=\min\{F(\omega),G(\omega)\}$, $(F\setminus G)(\omega)=\max\{F(\omega)-G(\omega),0\}$. Zaprojektuj map-reduce algorytm do wyznaczania tych trzech operacji. Algorytm na wejściu dostaje listę elementów zbioru

$$\{(1,\omega,F(\omega)) : \omega\in\Omega\wedge F(\omega)>0\}\cup\{(2,\omega,G(\omega)) : \omega\in\Omega\wedge G(\omega)>0\}$$

Zadanie 27 — W pliku `word-count.scala` znajduje się skrypt symulujący pracę systemu MapReduce dla problemu word-count.

1. Przekształć ten skrypt w bardzie realistyczny model - zapisz wynik pośredni (zmienna `keyval` z funkcji `TextMapper`) do pliku roboczego. Funkcja `TextReducer` ma pobierać wyniki z tego pliku.
2. Skróć przekształcony skrypt. Na przykład, dwie linijki z pliku `word-count.scala`

```
val groupped = keyval.groupBy(_._1)
val reduced  = groupped.mapValues(_.size)
```

mogą być skrócone do jednej linijki

```
val reduced = keyval.groupBy(_._1).mapValues(_.size)
```

4 Podobieństwo tekstów

Zadanie 28 — Pokaż, że funkcja $d(A,B) = |A \triangle B|$ jest metrykę na przestrzeni niepustych skończonych podzbiorów ustalonego zbioru X.

Zadanie 29 — Niech $f : [0,\infty) \to [0,\infty)$ będzie funkcją rosnącą i wklęsłą.

1. Pokaż, że dla $a, b \geq 0$ mamy $f(a+b) \leq f(a) + f(b)$.
   Wskazówka: Zauważ, że możemy założyć, że $a + b > 0$; następnie zauważ, że $a = (a+b)\frac{a}{a+b}$ oraz $b = (a+b)\frac{b}{a+b}$ i zastosuj nierówność Jensena dla funkcji wklęsłych.
2. Załóżmy dodatkowo, że $f(0) = 0$. Niech d będzie metryką na zbiorze X. Pokaż, że funkcja $\rho(x,y) = f(d(x,y))$ jest również metryką na zbiorze X.
3. Pokaż, że jeśli $\epsilon \in (0,1)$ oraz d jest metryką na zbiorze X, to funkcja $\rho(x,y) = d(x,y)^\epsilon$ jest metryką na zbiorze X.
4. Pokaż, że jeśli d jest metryką na zbiorze X, to funkcja $\rho(x,y) = \frac{d(x,y)}{1+d(x,y)}$ jest metryką na zbiorze X.

Zadanie 30 — Wybierzmy dwa losowe m-elementowe podzbiory A, B n elementowego zbioru X. Jaka jest wartość oczekiwana podobieństwa Jacckarda $J(A,B)$?

Zadanie 31 — Korzystając z Twierdzenia o Gęstości Liczb Pierwszych (Prime Numbers Theorem) oszacuj liczbę liczb pierwszych z przedziału $[2^{64}, 2^{64} + 1000]$ i następnie wyznacz te liczby.

Zadanie 32 — (Twierdzenie Steinhausa) Niech d będzie metryką na zbiorze X. Ustalmy element $a \in X$ i zdefiniujmy funkcję

$$\rho(x,y) = \frac{2d(x,y)}{d(x,a) + d(y,a) + d(x,y)}$$

Celem tego zadania jest pokazanie, że ρ jest metryką na zbiorze X.

1. Pokaż najpierw, że jeśli $0 < p \leq q$ oraz $r \geq 0$ to $\frac{p}{q} \leq \frac{p+r}{q+r}$.
2. Wprowadź oznaczenia $p = d(x,y)$, $q = d(x,y) + d(x,a) + d(y,a)$ oraz $r = d(x,z) + d(y,z) - d(x,y)$ i zastosuj obserwację z poprzedniego punktu do pokazania nierówności trójkąta dla funkcji ρ.

Zadanie 33 — Zastosuj Twierdzenie Steinhausa do metryki $d(X,Y) = |X \triangle Y|$ na zbiorze skończonych podzbiorów zbioru Ω do pokazania, że funkcja $d(X, Y0 = 1 - S(X,Y)$ (odległość Jaccarda) jest metryką.

Zadanie 34 — Załóżmy, że S jest takim podobieństwem obiektów przestrzeni Ω, że istnieje rodzina funkcji haszujących $\mathcal{H}$ oraz prawdopodobieństwo na rodzinie $\mathcal{H}$ takie, że dla dowolnych dwóch obiektów $A, B \in \Omega$ mamy

$$P_h[h(A) = h(B)] = S(A,B)$$

Pokaż, że wtedy funkcja $d(A,B) = 1 - S(A,B)$ jest metryką na zbiorze Ω.

Zadanie 35 — Uzupełnij szczegóły dowodu tego, że jeśli $\Omega = \{\omega_i : 1 \leq N\}$, π jest losową permutacją zbioru $\{1,\ldots,N\}$ (wybieraną zgodnie z rozkładem jednostajnym), oraz $h_\pi(X) = \min\{k : \omega_{\pi(k)} \in X\}$ dla $X \subseteq \Omega$ to

$$P_\pi[h_\pi(A) = h_\pi(B)] = S(A,B)\ .$$

Zadanie 36 — Napisz procedurę o specyfikacji `jaccard(f1:String,f2:String,k:Integer):Double`, która dla plików o nazwach `f1`, `f2` wyznacza ich `k`-gramy i następnie wylicza ich odległość Jaccarda. Przed wyznaczeniem k-gramów pliki powinne być oczyszczone (minimum to usunięcie znaków nowej linii, tabulatorów oraz podwójnych spacji)

1. Zastosuj tę procedurę do kilku wariantów swoich plików z algorytmami (zastosuj 4-gramy)
2. Zastosuj tę procedurę do porównania kolejnych rozdziałów analizowanej w Zadaniu 2 książki (zastosuj 7-gramy)

Zadanie 37 — Zastosuj metodę minhash do poprzedniego zadania. Twoja procedura powinna zależeć od parametru H który określa liczbę funkcji haszujących stosowanych do budowania sygnatury tekstu.

1. Przetestuj tę procedurę na danych z poprzedniego zadania hla $H \in \{50, 100, 250\}$ - porównaj aproksymację odległości Jaccarda z jej dokładnymi wartościami.

Pamiętaj o wygenerowaniu wspólnej rodziny funkcji haszujących dla wszystkich analizowanych tekstów.

Zadanie 38 — Napisz procedurę służącą do wyznaczania sygnatur kosinusowych plików tekstowych korzystających z 1024 losowych wektorów z $\mathbb{R}^n$ (n tutaj oznacza moc wspólnego zbioru słów występujących w badanych dokumentach). Dokumenty reprezentowane mają być przez wektor częstotliwości słów.

1. Zastosuj tę metodę do plików z Zadanie 2.

5 Streaming

Zadanie 39 — Niech C_n będzie wartością licznika Morrisa po n krotnym wywołaniu procedury INCREMENT. Niech $L_n = 2^{C_n}$.

1. Wyznacz wariancję zmiennej L_n oraz oblicz $\frac{\sigma(L_n)}{E(L)n)}$.
2. Zbadaj eksperymentalnie dokładność zastawu czterech liczników Morrisa. Jako estymator liczby n przyjmij $2^{(C_1(n)+C_2(n)+C_3(n)+C_4(n))/4}$.
3. Dla jakich n mamy $4\log_2(\log_2(n)) < \log_2(n)$?

Zadanie 40 — Zaimplementuj w Scali **Boyer–Moore'a Majority Algorithm**.

1. Napisz najpierw funkcję której parametrem jest lista łańcuchów (List[String]).
2. Zaprojektuj następnie obiekt o dwóch metodach: add(x:String):Unit oraz get():String który realizuje ten algorytm.
3. Jaka jest złożoność obliczeniowa i pamięciowa tego algorytmu.

Zadanie 41 — Zaimplementuj w Scali **Misra - Gries Algorithm**.

1. Napisz najpierw funkcję której parametrami jest lista łańcuchów (List[String]) oraz liczba k określającą maksymalną liczbę śledzonych obiektów. Skorzystaj z kolekcji scala.collection.mutable.Map
2. Zaprojektuj następnie obiekt o dwóch metodach: add(x:String):Unit oraz get():String który realizuje ten algorytm. Do utworzenia tego obiektu potrzebujesz jeden parametr k.
3. Jaka jest złożoność obliczeniowa i pamięciowa tego algorytmu.

Zadanie 42 — Algorytm HyperLogLog używa wartości $h(x) = (b_0 b_1 b_2 b_3 \ldots)$ do wyznaczenia numeru inkrementowanego licznika (i = $(b_0 \ldots b_{k-1})_2 + 1$) oraz do wyznaczenia z reszty ciągu bitów $(b_k b_{k+1} \ldots)$ do zwiększenia wartości licznika. Załóżmy, że $h(x)$ jest typy Int lub Long oraz, że $h(x) \geq 0$.

1. Jak można za pomocą operacji bitowych wyznaczyć z $h(x)$ ciąg $(b_0 \ldots b_{k-1})$?
2. Jak można za pomocą operacji bitowych wyznaczyć z $h(x)$ ciąg $(b_k k_{k+1} \ldots)$?
3. Załóżmy, że $n \geq 0$. Co robi operacja $n\&(-n)$. Jak tą operację możesz wykorzystać do inkrementacji licznika.

Zadanie 43 — Zaimplementuj w Scali algorytm **HyperLogLog**.

1. Pobierz ze strony http://ita.ee.lbl.gov/html/contrib/LBL-PKT.html plik lbl-pkt-4. Wypakuj z niego plik lbl-pkt-4.tcp. Oto format danych: timestamp, (przenumerowany) source host, (przenumerowany) destination host, source TCP port, destination TCP port, liczba bajtów danych(zero dla "pure-ack"pakietów).
2. Zastosuj HupeLogLog do wyznaczenia liczby różnych source hostów, liczby różnych destination hostów oraz liczby różnych par (source,destination).

Zadanie 44 — Pokaż, że wielomian $w(x) = 1 + x + x^2$ jest nierozkładalny w pierścieniu $\mathbb{Z}_2[x]$. Rozważ ideał $(w) = \{\alpha \cdot w : \alpha \in \mathbb{Z}_2[x]\}$ w pierścieniu $\mathbb{Z}_2[x]$. Obliczenia będziemy prowadzić w pierścieniu ilorazowym $\mathbb{Z}_2[x]/(w)$.

1. Pokaż, że dla $\alpha, \beta \in \mathbb{Z}_2[x]$ mamy

$$((w) + \alpha = (w) + \beta) \equiv w|(\alpha - \beta)$$

2. Pokaż, że struktura $(\mathbb{Z}_2[x]/(w), +)$ jest izomorficzna z $(\mathbb{Z}_2^2, +)$.
3. Oznacz przez i zmienną x. Zauważ, że $\mathbb{Z}_2[x]/(w) = \{[0], [1], [i], [1+i]\}$.

4. Stosując oznaczenia: $\mathbf{0} = [0]$, $\mathbf{1} = [1]$, $\mathbf{i} = [i]$ oraz $\mathbf{1} + \mathbf{i} = [1+i]$ wyznacz tabliczki dodawania i mnożenia w $\mathbb{Z}_2[x]/(w)$
5. Pokaż, że $\mathbf{1} + \mathbf{i} + \mathbf{i^2} = \mathbf{0}$.
6. Pokaż, że $\mathbb{Z}_2[x]/(w)$ jest czteroelementowym ciałem.

Zadanie 45 — Powtórz poprzednie zadanie dla wielomianu $w(x) = 1+x+x^3$. Skonstruujesz w ten sposób ciało 8 elementowe.

Zadanie 46 — Zbuduj ciało 9 elementowe.

Zadanie 47 — Niech F będzie ciałem. Niech $a_1, \ldots, a_4$ będą parami różnymi elementami ciała F. Niech $b_1, \ldots, b_4$ będą dowolnym elementami ciała F. Niech

$$w(x) = \sum_{i=1}^{4} b_j \prod_{j \neq i} \frac{x - a_j}{a_i - a_j}$$

(jest to wielomian interpolacyjny Legrange'a stopnia 4). Pokaż, że w jest jednym wielomianem stopnia trzeciego z pierścienia $F[x]$ takim, że $w(a_1) = b_1$, $w(a_2) = b_2$, $w(a_3) = b_3$ oraz $w(a_4) = b_4$.

Zadanie 48 — Wygeneruj listę $\{h_1, \ldots, h_{100}\}$ losowych wielomianów stopnia 3 nad ciałem $\mathbb{Z}_{11}$.
Wskazówka: W programie Mathematica można to zrobić za pomocą następującej funkcji:
LP[p_]:= Module[{},Function[x,Mod[RandomInteger[{0,p-1},4].$\{1, x, x^2, x^3\}$,p]]]; .

1. Wyznacz moc zbioru $\{i \in \{1, \ldots, 100\} : h_i(1) = 2\}$. Powtórz ten eksperyment kilka razy. Pamiętaj aby obliczenia wykonywać w ciele $\mathbb{Z}_{11}$.
2. Wygeneruj histogram wartości $\{h_i(1) : i \in \{1, \ldots, 100\}\}$.

Zadanie 49 — Zapoznaj się z testem nierozkładalności wielomianów Rabina i pokaż, że wielomian $w(x) = x^{80} + x^9 + x^4 + x^2 + 1$ jest wielomianem nierozkładalnym nad ciałem $\mathbb{Z}_2$. Oblicz $[x^{40}] \cdot [x^{40}]$ w ciele $\mathbb{Z}_2[x]/(w)$.

Zadanie 50 — Oprogramuj w języku Scala Geometric Histrogram Streaming Window Algorithm M. Datara, A. Gionisa, P. Indyka i R. Motwaniego z pracy `http://www-cs-students.stanford.edu/~datar/papers/sicomp_streams.pdf`

6 Page Rank

Zadanie 51 — Niech $A = (a_{ij})$ będzie kwadratową macierzą rozmiaru $n \times n$. Niech $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Pokaż, że

$$||Ax^T||_2 \leq \sqrt{\sum_{i,j} a_{ij}^2} \cdot ||x||_2 \ .$$

(gdzie $||(y_1, \ldots, y_n)||_2 = \left(\sum_i x_i^2\right)^{1/2}$). Wywnioskuj z tego, że jeśli λ jest wartością własną macierzy kolumnowo stochastycznej, to $|\lambda| \leq 1$.

Zadanie 52 — Załóżmy, że A jest macierzą kwadratową o współczynnikach rzeczywistych. Pokaż, że jeśli λ jest wartością własną macierzy A, to liczba $\overline{\lambda}$ (sprzężenie liczby λ) jest również wartością własną macierzy A.

Zadanie 53 — Załóżmy, że $A, B \in M_n(\mathbb{R})$ są kolumnowo stochastyczne.

1. Pokaż, że macierz $A \circ B$ jest kolumnowo stochastyczna.
2. Niech $\alpha \in [0, 1]$. Pokaż, że macierz $\alpha A + (1 - \alpha)B$ jest kolumnowo stochastyczna.
3. Podaj interpretacje probabilistyczne powyższych faktów.

Zadanie 54 — Załóżmy, że w grafie nie ma 'wiszących wierzchołków'. Niech v będzie wierzchołkiem bez linków do tego wierzchołka. Pokaż, że $pagerank(v) = \frac{1-\alpha}{n}$.

Zadanie 55 — Rozważmy gwiazdę rozmiaru $n+1$, czyli graf o krawędziach $\{i \to n+1, i = 1 \ldots, n\}$. Wyznacz PageRank dla tego grafu.

Zadanie 56 — Wyznacz PageRank dla grafu $\{a \to b, b \to a, b \to c\}$.

Zadanie 57 — Zainstaluj pakiet Jama (załaduj ze strony http://math.nist.gov/javanumerics/jama/Jama-1.0.3.jar plik do katalogu scala/lib).

1. Napisz procedurę służącą do wygenerowania macierzy Google dla porządku liniowego rozmiaru n.
2. Napisz procedurę 'naiwnego' obliczania PageRank metodą potęgową i zastosuj ją wyznaczenia PageRank dla tego liniowego porządku.

Zadanie 58 — Przeskanuj witrynę WWW któregoś z pracowników katedry. Wyznacz PageRank dla wszystkich wierzchołków tego grafu.

Zadanie 59 — Napisz pseudo-kod funkcji MAPPER i REDUCER służących do wykonania jednego kroku iteracyjnego wyznaczenia PageRank metodą polegającą na rozbiciu macierzy przejść M na k^2 bloków:

$$\begin{bmatrix} M_{11} & \dots & M_{1k} \\ \vdots & \vdots & \vdots \\ M_{k1} & \dots & M_{kk} \end{bmatrix} \circ \begin{bmatrix} V_1 \\ \vdots \\ V_k \end{bmatrix} = \begin{bmatrix} M_{11} \circ V_1 + \ldots + M_{1k} \circ Vk \\ \vdots \\ M_{k1} \circ V_1 + \ldots + M_{kk} \circ Vk \end{bmatrix}$$

1. Pokaż poprawność metody mnożenia macierzy przez wektor za pomocą podziału na bloki.
2. Niech R będzie pierścieniem łącznym. Pokaż, że

$$(M_{n\times n}(M_{k\times k}(R)), \circ) \cong (M_{(nk)\times(nk)}(R), \circ)$$

(gdzie $M_{m\times m}(R)$ oznacza zbiór macierzy kwadratowych wymiaru $m \times m$ o wyrazach z pierścienia R).

Zadanie 60 — Wybierz witrynę jednego z pracowników Katedry Informatyki. Sprawdź jej poprawność za pomocą narzędzi ze stron

•https://validator.w3.org/,

•https://jigsaw.w3.org/css-validator/

•https://search.google.com/search-console/mobile-friendly.

Sprawdź następnie poprawność meta-informacji i strukturę semantyczną głównej strony.

Zadanie 61 — Niech L oznacza zbiór wszystkich liniowych porządków na zbiorze $X = \{a, b\}$. Ile jest funkcji $F : L^n \to L$ spełniających zasadę Pareto?

Zadanie 62 — Niech L oznacza zbiór wszystkich liniowych porządków na zbiorze $X = \{a, b, c\}$. Ile jest funkcji $F : L^n \to L$ spełniających zasadę Pareto oraz zasadę uczciwości (niewrażliwości na trzecią możliwość)?

7 Frequent itemsets

Zadanie 63 — Załóżmy, że zbiór obiektów I ma d elementów.

1. Pokaż, że moc zbioru wszystkich reguł $X \Rightarrow Y$ takich, że $X, Y \subseteq I$ wynosi $3^d - 2^{d+1} + 1$.
2. Ile jest reguł postaci $X \Rightarrow \{a\}$?
3. Ile jest reguł postaci $\{a, b\} \Rightarrow \{c\}$?

Zadanie 64 — Niech $I = \{1, \ldots, 100\}$ oraz $T_b = \{x \in I : x|b\}$ dla $b = 1, \ldots, 100$.

1. Wyznacz częste obiekty przyjmując z próg supportu liczbę α=5%.
2. Które pary są częste dla tego samego progu α ?
3. Wyznacz moc koszyka T_b oraz sumę $\sum_{b=1}^{100} |T_b|$
4. Jakie są współczynniki wiarygodności reguł $\{5, 7\} \Rightarrow \{2\}$ oraz $\{2, 3, 4\} \Rightarrow \{5\}$? Jakie są ich współczynniki wzmocnienia (liftingu)?

Zadanie 65 — Niech $I = \{1, \ldots, 100\}$ oraz $T_b = \{x \in I : b|x\}$ dla $b = 1, \ldots, 100$. Odpowiedz na te same pytania co w poprzednim zadaniu.

Zadanie 66 — (Scala) Niech T = $[T_1, T_2, \ldots, T_n]$ będzie tablicą zbiorów łańcuchow (typu Array[Set[String]]).

1. Wyznacz zbiór $T_1 \cup \ldots T_n$ za pomocą metody reduce.
2. Przekształć otrzymany zbiór łańcuchów w listę za pomocą odpowiedzniej metody klasy Set
3. Zastosuj metodę zipWithIndex i sprawdź otrzymany obiekt
4. Przekształć otrzymany obiekt w obiekt o nazwie S2I typu Map[String,Int].
5. Zapisz ciąg operacji z punktów 1,2,3,4 za pomocą jednej linii kodu
6. Odwróć obiekt S2I, tzn zbuduj obiekt I2S typu Map[Int,String] taki, że

$$((i \to s) \in I2S) \longleftrightarrow ((s \to i) \in S2I)$$

Zadanie 67 — (Scala) Masz daną tablicę transakcji T = $[T_1, T_2, \ldots, T_n]$. Zastosuj poprzednie zadanie jako metodą na zbudowanie słownika, który jednoznacznie numeruje łańcuchy występujące w elementach T liczbami naturalne od 0 do pewnej liczby $m - 1$

1. Przekształć tablicę T w tablicę TI, w której wszystkie łańcuchy są zastąpione odpowiadających im liczbą.
2. Wyznacz mapę RCR = $\{(i, s_i), i = 0, \ldots m\}$, gdzie $s_i = s(\{i\})$
3. Przefiltruj mapę C pozostawiając w niej te elementy i, że $s_i \geq 0.1 \cdot n$. Zastosuj do tego celu metodę retain.

4. Zbuduj zbiór częstych obiektów (wykorzystaj do tego celu przefiltrowaną mapę C)

Zadanie 68 — Zbiór X nazywamy maksymalnym jeśli $s(X) \geq s_{min} \cdot n$ oraz dla dowolnego $Y \supset X$ mamy $s(Y) < s_{min} \cdot n$. Pokaż, że dla dowolnego częstego zbioru X istnieje maksymalny zbior M taki, że $X \subseteq M$.

Zadanie 69 — Zbiór X nazywamy domkniętym jeśli $s(X) \geq s_{min} \cdot n$ oraz dla dowolnego $Y \supset X$ mamy $s(Y) < s(X)$.

1. Pokaż, że dla dowolnego częstego zbioru X istnieje zbior domknięty C taki, że $X \subseteq C$.
2. Niech X będzie częsty. Pokaż, że

$$s(X) = \max\{s(C) : X \subseteq C \land C \text{ jest domknięty}\} \ .$$

Zadanie 70 — Pobierz program Apriori Christiana Borgelt'a. Zapoznaj się z parametrami wywołania tego programu. Przyjrzyj się interesującym regułom ze zbioru votes.txt (znajdziesz go na stronie wykładu).

Zadanie 71 — Znajdź kilka interesujących reguł w zbiorze sumermarket.arff (znajdziesz go na stronie wykładu).

8 Klasteryzacja

Zadanie 72 — Niech X, Y będą losowymi punktami niezależnie wybranymi z odcinka $[0, 1]$ (zgodnie z rozkładem jednostajnym). Pokaż, że $E(|X - Y|) = \frac{1}{3}$.

Zadanie 73 — Widząc, że objętość n-wymiarowej kuli o promieniu r wynosi $\frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} r^n$ wyznacz objętość elipsy

$$\{x \in \mathbb{R}^n : \sum_i^n \left(\frac{x_i}{a_i}\right)^2 \leq 1\} \ .$$

Zadanie 74 — Niech $X_1, \ldots, X_n$ oraz $Y_1, \ldots, Y_n$ będą niezależnymi zmiennymi losowymi o wartościach w zbiorze $\{-1, 1\}$ (rozważamy rozkład jednostajny). Wyznacz rozkład zmiennej losowej $\sum_{i=1}^n X_i Y_i$.

Zadanie 75 — Wykonaj procedurę hierarchicznej klasteryzacji jednowymiarowego zbioru punktów {1, 4, 9, 16, 25, 36, 49, 64, 81}, stosując następujące metody łączenia klastrów:

1. klastry są reprezentowane przez ich centroidy (średnia wartość); w każdym kroku łączone są klastry z najbliższymi centroidami
2. stosujemy metodę *single link*, $d(C, D) = \min\{d(x, y) : x \in C, y \in D\}$

3. łączymy klastry (C, D) minimalizujące *promień* klastra $C \cup D$; uwaga: przez promień zbioru X rozumiemy liczbę
$$r(X) = \min_{x \in X} \max_{y \in X} d(x, y)$$

4. łączymy klastry (C, D) minimalizujące *średnicę* klastra $C \cup D$; uwaga: przez średnicę zbioru X rozumiemy liczbę
$$\Delta(X) = \max_{x, y \in X} d(x, y)$$

5. Pokaż, że $\Delta(X) \leq 2r(X)$.

c.d.n.
Powodzenia,
Jacek Cichoń