アルゴリズムとデータ構造III
13回目：1月7日（木）

暗号，符号化，テキスト圧縮

授業資料　http://ir.cs.yamanashi.ac.jp/~ysuzuki/algorithm3/

授業の予定（中間試験まで）

1	10/01	スタック（後置記法で書かれた式の計算）
2	10/15	文脈自由文法，構文解析，CYK法
3	10/22	構文解析　CYK法
4	10/29	構文解析　CYK法
5	11/12	構文解析　CYK法，動的計画法
6	11/19	構文解析（チャート法），グラフ（ダイクストラ法）
7	11/26	グラフ（ダイクストラ法，DPマッチング，A*アルゴリズム）
8	12/03	グラフ（A＊アルゴリズム），前半のまとめ
9	12/04 4時限	教室：A1-41 全文検索アルゴリズム（simple search, KMP）

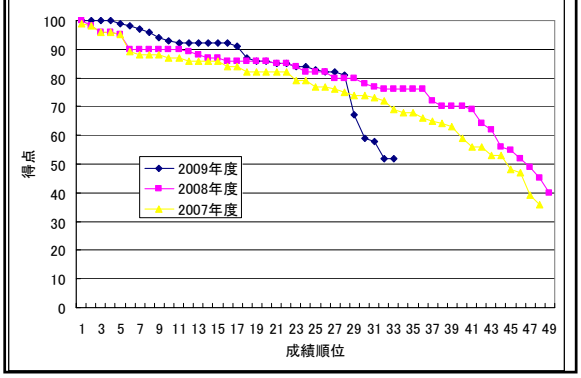
授業の予定（中間試験以降）

10	12/10	中間試験（8回目までの範囲）
11	12/11 4時限	教室：A1-41 全文検索アルゴリズム（BM, Aho-Corasick)
12	12/17	全文検索アルゴリズム（Aho-Corasick），データ圧縮
13	01/07	暗号（黄金虫，踊る人形） 符号化（モールス信号，Zipfの法則，ハフマン符号）テキスト圧縮
14	01/14	テキスト圧縮　（zip）， 音声圧縮　（ADPCM，MP3，CELP）， 画像圧縮（JPEG）
15	02/04	**期末試験**

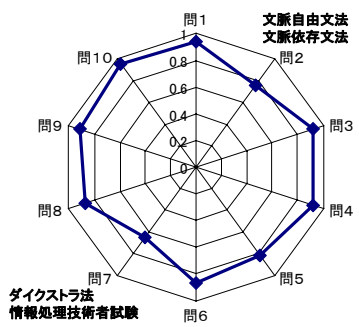
中間試験の結果

	今年 09年度	08年度	07年度
受験者	33人	49人	48人
平均点	86点	79点	75点
満点獲得者数	4人	1人	0人

中間試験の結果



問題別得点結果



中間試験の解答例

http://ir.cs.yamanashi.ac.jp/~ysuzuki/algorithm3/
の12月10日の授業資料

本日のメニュー

- Zipfの法則
- 暗号
  - 黄金虫(The gold bug)
  - 踊る人形(The Adventure of the Dancing Men)
- 符号化
  - モールス信号
  - ハフマン符号
- テキスト圧縮

ジップの法則(Zipf's law)

「あるタイプの現象が生起する確率はその現象の生起する順位に反比例する」:経験則

$$\text{生起確率} = \frac{\text{定数}C}{\text{順位}}$$

- Zipfの法則が当てはまる事象
  - 文字毎の出現頻度
  - コンピュータにおけるコマンドの使用頻度
  - Webページのアクセス頻度
  - 都市の人口
  - 文献の参照回数
  - 会社でのランク(役職)と給料など
  - ケータイのシェア(docomo, au, softbank, e-mobile)

携帯電話:各グループ毎の加入者数累計(2009年12月 ケータイWatchより)

順位	事業者	累計	割合(確率)	Zipf's law C=0.51
1	NTTドコモ	55,297,200	50.2%	51.0%
2	KDDI	31,329,400	28.4%	25.5%
3	ソフトバンク	21,501,900	19.5%	17.0%
4	イー・モバイル	2,048,200	1.8%	12.8%

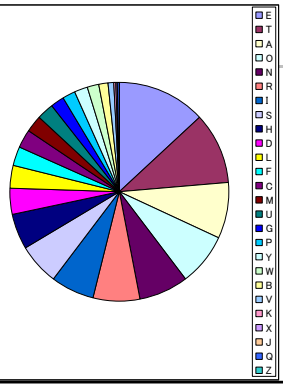
$$\text{生起確率} = \frac{\text{定数}C}{\text{順位}}$$

自然言語の統計的性質

- 文字の使用頻度(英語) _はスペース

順位	文字	%	2	%	3	%	4	%
1	_	17.4	e_	3.0	_th	1.6	_the	1.2
2	e	9.7	_t	2.4	the	1.3	the_	1.0
3	t	7.0	th	2.0	he_	1.3	_of_	0.6
4	a	6.1	he	1.9	_of	0.6	and_	0.4
5	o	5.9	_a	1.7	of_	0.6	_and	0.4
6	i	5.5	s_	1.7	ed_	0.5	_to_	0.4
7	n	5.5	d_	1.5	_an	0.5	ing_	0.3

文字の使用頻度(caesarより)



順位	文字	出現確率	順位	文字	出現確率
1	E	0.1300	14	M	0.0250
2	T	0.1050	15	U	0.0240
3	A	0.0810	16	G	0.0200
4	O	0.0790	17	P	0.0190
5	N	0.0710	18	Y	0.0190
6	R	0.0680	19	W	0.0150
7	I	0.0630	20	B	0.0140
8	S	0.0610	21	V	0.0090
9	H	0.0520	22	K	0.0040
10	D	0.0380	23	X	0.0015
11	L	0.0340	24	J	0.0013
12	F	0.0290	25	Q	0.0011
13	C	0.0270	26	Z	0.0007

単語の使用頻度

順位	単語	%	2	%	3	%
1	the	6.1	of the	0.9	one of the	0.03
2	of	3.5	in the	0.5	as well as	0.02
3	and	2.7	to the	0.3	the United States	0.02
4	to	2.5	on the	0.2	out of the	0.02
5	a	2.1	and the	0.2	some of the	0.01
6	in	1.9	for the	0.1	the end of	0.01
7	that	0.9	to be	0.1	the fact that	0.01

単語の出現頻度分布

- ジップの法則(Zipf's law):
  - 単語の出現順位(r)と出現頻度(f)は反比例の関係にある

$$r = \frac{C}{f} \qquad f = \frac{C}{r}$$

n番目の単語の出現確率 P_n

$$P_n = \frac{C}{n}$$

順位	文字	出現確率	0.065/順位
1	the	0.061	0.065
2	of	0.035	0.0325
3	and	0.027	0.0108333
4	to	0.025	0.0027083
5	a	0.021	0.0005417
6	in	0.019	0.00009028
7	that	0.009	0.0000129

Cは定数
低頻度の語には当てはまらない

データの頻度分布の偏りを利用した技術

- 暗号(換字式)の解読
  - 小説(ポー, ドイルなど)
  - Code talker (戦時中の暗号通信兵　米映Windtalkers)
- データ圧縮(ロスレス)
  - キー入力時の打鍵回数の削減
  - モールス符号
  - ハフマン符号(情報理論　2年前期　宮本先生)
- Boyer-Moore法(全文検索アルゴリズム)
  - キーワードに含まれない文字を積極的に利用

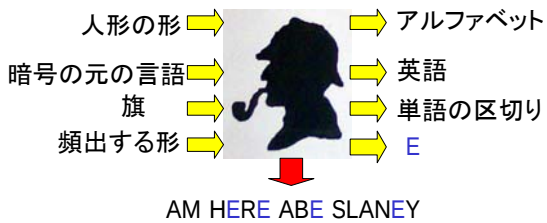
小説中での暗号解読の解説

- 黄金虫(The gold bug)
  - 著者:エドガー・アラン・ポー
  - 作品:翻訳版
    - http://www.aozora.gr.jp/cards/000094/card2525.html
  - 作品:原文
    - http://en.wikisource.org/wiki/The_Gold-Bug
- 踊る人形(The Adventure of the Dancing Men)
  - 著者:アーサー・コナン・ドイル
  - 作品:翻訳版　題:暗号舞踏人の謎
    - http://www.aozora.gr.jp/cards/000009/card45340.html
  - 作品:原文
    - http://en.wikisource.org/wiki/The_Adventure_of_the_Dancing_Men

黄金虫(エドガー・アラン・ポー)に出てくる暗号(換字式)

- 小説内で暗号解読
- 暗号は多分英語
- 英語は文字によって出現確率が違う
  - 出現確率の高い方から並べると
    - e a o i d h n r s t u y c f g l m w b k p q x z (eは頻出)
  - eeも頻出
  - theも頻出
- 対応がとれた文字は置き換え，前後の文字を推理する

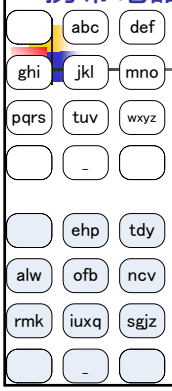
「踊る人形」アーサー・コナン・ドイル (The Adventure of the Dancing Men)



AM HERE ABE SLANEY


"What one man can invent another can discover."

携帯電話のアルファベットキー



- 一般的なアルファベットキー
  - アルファベット順に26文字を8つのキーに割り振っている
  - pqrsとwxyzは4文字を1つのキーに割り振られている

キー配置による打鍵数の違い

	i		h	a	v	e		a		p	e	n	合計
上	3	1	2	1	3	2	1	1	1	1	2	2	20
下	1	1	2	1	3	1	1	1	1	3	1	1	17

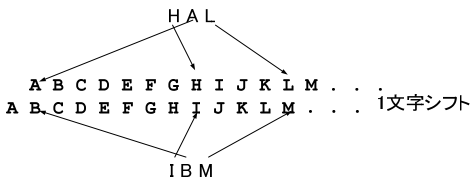
- 出現頻度を考慮したアルファベットキー（鈴木考案）
  - 出現頻度が低い文字を入力するには複数回打鍵
  - キーの場所を覚え直す必要

おまけ
Scrabble (英単語作成ボードゲーム)の得点

- Scrabble
  - 対戦型英単語作成ゲーム
    - ボード上に手持ちの文字をならべ英単語を作成
    - 作成した単語の文字に書かれている得点を合計し，高得点を競う
  - 英単語を作りにくい文字には高得点が割り振られている．
    - 1点：E, A, I, O, R, N, T, L, S, U
    - . . . . . .
    - 10点：Q, Z

シフト暗号（蛇足）

- シーザー暗号
- ROT13, ROT47
- 「2001年宇宙の旅」のHAL ← IBM （俗説？）



Caesar（シーザー式暗号法の解読）

- Unixのアプリケーション
- kkiではオンラインマニュアルはあるがプログラム自身はインストールされていない
- CentOSやUbuntuでは（インストールすれば）使用可能（のはず）

使用例
>caesar
J ibwf b qfo ← I have a pen を1文字ずらして入力
>I have a pen ← 各文字の出現頻度を利用し，何文字ずらしたかを推測し答えを出力する

Code talker（暗号通信兵）

- Windtalkers (アメリカ映画　2002年)
  - アメリカインディアンのナバホ族が暗号通信兵
  - ナバホ族の言葉を使って暗号通信
  - サイパン島での日本軍との戦い
  - ナバホ族の言葉
    - 文法も発音も独特（nativeにしか理解できない）
    - 日本軍は知らない
    - アメリカにはnativeのナバホ族がいる（訓練しなくても理解できる）

頻度分布の偏りのデータ圧縮への利用

- モールス信号
- ハフマン符号

モールス信号の符号

- ・(短点)と−(長点)を用いてアルファベットを表現する
- 情報を早く送るための工夫
  - よく使われる文字(例えばe,t)は短い
    - e: ・ (短点1文字)
    - t: − (長点1文字)
  - あまり使われない文字(例えばqは4文字)は長い
    - q: −−・−

モールス信号の符号

- ・(短点)と−(長点:短点3つ分の長さ)を用いてアルファベットを表現する
- 区切り記号
  - 文字の切れ目:短点3つ分の間隔
  - 単語の切れ目:短点7つ分の間隔
- L: ・−・・ (LifeカードのCMに使われていた)
- SOS:・・・ −−− ・・・

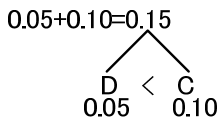
ハフマン符号

- 2分木を使って文字の出現頻度順に並べる
- 葉＝文字
- 浅い:符号長が短い, 深い:符号長が長い
- 平均符号長が最小になることが保証されている

ハフマン符号の作り方 1/5

文字	頻度
A	0.25
B	0.20
C	0.10
D	0.05
E	0.40

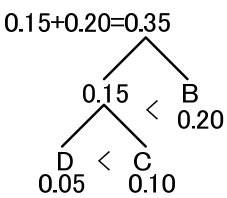
- 頻度の低い文字を2文字(DC)を選び, 頻度の低い方を左の葉, 頻度の高い方を右の葉に置き, 2分木をつくる.
- ルートノードには2つの葉の頻度の和を書き込む



ハフマン符号の作り方 2/5

文字	頻度
A	0.25
B	0.20
(DC)	0.15
E	0.40

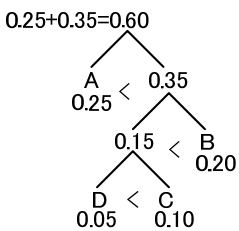
- (DC)統合後, 頻度の低いBと(DC)連合を選ぶ. Bと(DC)連合の頻度を比較し, 頻度の高いBを右ノードに, 低い(DC)連合を左ノードに配置する.
- ルートノードには頻度の和を書き込む



ハフマン符号の作り方 3/5

文字	頻度
A	0.25
((DC)B)	0.35
E	0.40

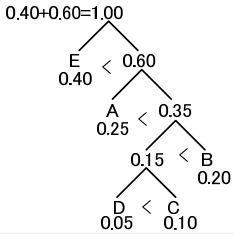
- ((DC)B)統合後, 頻度の低いAと((DC)B)連合を選ぶ. Aと((DC)B) 連合の頻度を比較し, 頻度の高い((DC)B) 連合を右ノードに, 低いAを左ノードに配置する.
- ルートノードには頻度の和を書き込む



ハフマン符号の作り方 4/5

文字	頻度
(A((DC)B))	0.60
E	0.40

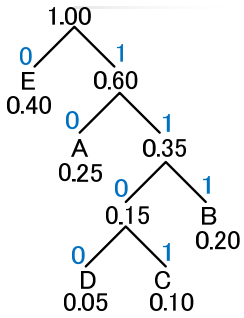
- (A((CD)B))統合後，頻度の低いEと(A((CD)B))連合を選ぶ．Eと(A((CD)B)) 連合の頻度を比較し，頻度の高い(A((CD)B))連合を右ノードに，低いEを左ノードに配置する．
- ルートノードには頻度の和を書き込む



ハフマン符号の作り方 5/5

- 左のノードに0，
  右のノードに1を付与する

文字	頻度	符号
A	0.25	10
B	0.20	111
C	0.10	1101
D	0.05	1100
E	0.40	0



文字⇔ハフマン符号の変換

文字	頻度	符号
A	0.25	10
B	0.20	111
C	0.10	1101
D	0.05	1100
E	0.40	0

- 10111110111000
  - 10|111|1101|1100|0
  - A|B|C|D|E
- BBCEDA
  - 11111111010110010
  - 111|111|1101|0|1100|10

ASCII文字コード(8bit)からハフマン符号へ

文字	頻度	符号
A	0.25	10
B	0.20	111
C	0.10	1101
D	0.05	1100
E	0.40	0

- A
- ASCII: 01000001 (0x41) 8bit
- Huffman: 10 : 2bit
- E
- ACSII: 01000101 (0x45) 8bit
- Huffman: 0 : 1bit

練習問題1

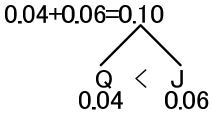
- 下の表のような記号の出現頻度のとき，ハフマン符号をつくりなさい．但しハフマン符号作成のための二分木も書くこと．

記号	頻度
B	0.17
D	0.20
E	0.33
F	0.12
J	0.06
K	0.08
Q	0.04
合計	1.00

練習問題1 解答例 1/7

- 下の表のような記号の出現頻度のとき，ハフマン符号をつくりなさい．但しハフマン符号作成のための二分木も書くこと．

記号	頻度
B	0.17
D	0.20
E	0.33
F	0.12
J	0.06
K	0.08
Q	0.04
合計	1.00



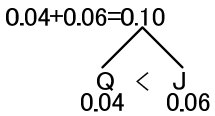
練習問題1 解答例 1/7

- 下の表のような記号の出現頻度のとき，ハフマン符号をつくりなさい．但しハフマン符号作成のための二分木も書くこと．

記号	頻度
B	0.17
D	0.20
E	0.33
F	0.12
J	0.06
K	0.08
Q	0.04
合計	1.00

⇒

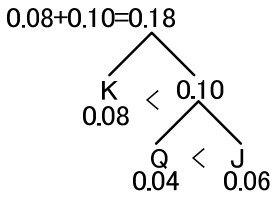
記号	頻度
E	0.33
D	0.20
B	0.17
F	0.12
K	0.08
J	0.06
Q	0.04
合計	1.00



練習問題1 解答例 2/7

- 下の表のような記号の出現頻度のとき，ハフマン符号をつくりなさい．但しハフマン符号作成のための二分木も書くこと．

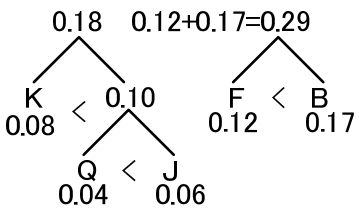
記号	頻度
E	0.33
D	0.20
B	0.17
F	0.12
(Q J)	0.10
K	0.08
合計	1.00



練習問題1 解答例 3/7

- 下の表のような記号の出現頻度のとき，ハフマン符号をつくりなさい．但しハフマン符号作成のための二分木も書くこと．

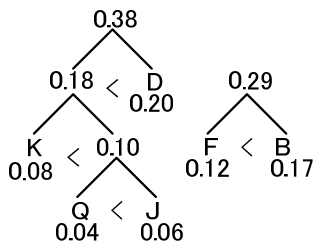
記号	頻度
E	0.33
D	0.20
(K(Q J))	0.18
B	0.17
F	0.12
合計	1.00



練習問題1 解答例 4/7

- 下の表のような記号の出現頻度のとき，ハフマン符号をつくりなさい．但しハフマン符号作成のための二分木も書くこと．

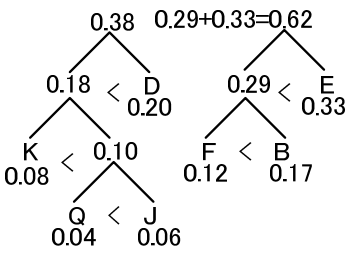
記号	頻度
E	0.33
(F B)	0.29
D	0.20
(K(Q J))	0.18
合計	1.00



練習問題1 解答例 5/7

- 下の表のような記号の出現頻度のとき，ハフマン符号をつくりなさい．但しハフマン符号作成のための二分木も書くこと．

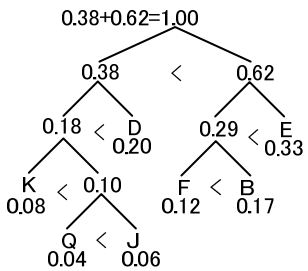
記号	頻度
((K(Q J))D)	0.38
E	0.33
(F B)	0.29
合計	1.00



練習問題1 解答例 6/7

- 下の表のような記号の出現頻度のとき，ハフマン符号をつくりなさい．但しハフマン符号作成のための二分木も書くこと．

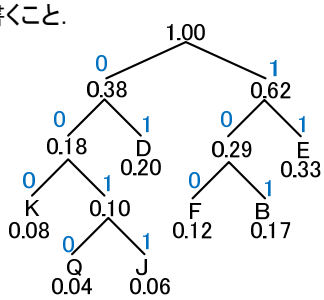
記号	頻度
((F B) E)	0.62
((K(Q J))D)	0.38
合計	1.00



練習問題1 解答例 7/7

- 下の表のような記号の出現頻度のとき，ハフマン符号をつくりなさい．但しハフマン符号作成のための二分木も書くこと．

記号	頻度	コード
B	0.17	101
D	0.20	01
E	0.33	11
F	0.12	100
J	0.06	0011
K	0.08	000
Q	0.04	0010
合計	1.00	



ハフマン符号の特徴

- 各記号がリーフノード（葉）に対応している
  - ハフマン符号列を左からトレースすることで，記号の区切りが分かる
  - 区切り記号を入れる必要がない

レポート

- Boyer-Moore法のプログラムを作成
  - 言語は何でも良い
  - プログラムの説明
- データ
  - text:
    - 1: ABCDABABCDEABCD
    - 2: ZYXWVUTSABCDEFG
  - Key:
    - 1: AB
    - 2: ABCD
- 結果表示（4種類の実験に対して）
  - キーワード出現位置（あれば複数）
  - 照合回数
- 締め切り：2月12日（金）17:00
- 提出場所：鈴木の居室前のレポート入れ