アルゴリズムとデータ構造

第4回 集合の基本操作

http://www-ui.is.s.u-tokyo.ac.jp/~takeo/course/algorithm/

五十嵐 健夫

takeo@is.s.u-tokyo.ac.jp

前回の内容

木の定義
木の実現
2分木
ハフマンコード

今回の内容

集合
辞書
ハッシュ
- オープンとクローズド
- 効率の解析

優先度付待ち行列
複雑な集合構造

集合

要素の集まり。重複なし。

線形順序、空、包含、和、積、差

union(A,B), intersection(A,B), difference(A,B)
member(x), makenull(), insert(x), delete(x),
merge(A,B), assign(A,B)
min(), equal(A,B), find(x)

集合の実現

ビットベクトルによる表現 (有限個)

Unionの例

連結リストによる表現

ソート済みの場合

Intersectionの例

辞書

insert, delete, member, makenull のみの集合

(賛成・反対議員の例)

リスト、ビットベクトル、配列…

ハッシュ表

ハッシュ表

辞書引きを平均O(1)で実現

大きさ制限なし:オープンハッシュ

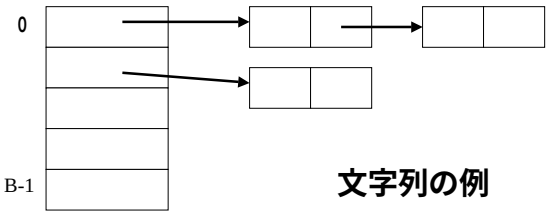
ハッシュ関数、バケット表、要素リスト

大きさ制限あり: クローズドハッシュ

再ハッシュ

大きさ制限なし：オープンハッシュ

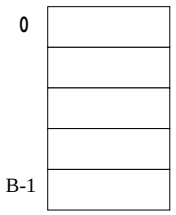
ハッシュ関数h、バケット表、要素リスト



文字列の例

大きさ制限あり：クローズドハッシュ

再ハッシュ、“deleted”**マーク**



一次ハッシュ法

ハッシュ関数の効率の推定

オープンハッシュでの一回操作 O(1+N/B)

文字コードを使ったハッシュの効率

平方採中法 $h(n) = \lfloor n^2 / C \rfloor \bmod B$

$BC^2 \approx K^2$

文字列の場合

クローズドハッシュ法の解析

等確率にふさがっていて、ランダムに割り当てる場合の操作時間

i回以上衝突の確率 $(N/B)^i$

調べるバケットの平均個数 $\frac{B}{B-N}$

N=B/2**なら平均2個**

M個のデータを入れる平均の手間 $\frac{B}{M}\log_e \frac{B}{B-M+1}$

クローズドハッシュ法の解析

等確率にふさがっていて、ランダムに割り当てる場合の操作時間

埋まっている割合（α）に応じた操作時間

挿入時間、非存在要素の検索時間 $\frac{1}{1-\alpha}$

削除、存在要素の検索時間 $-\frac{1}{\alpha}\log_e(1-\alpha)$

クローズドハッシュ法の解析

等確率にふさがっていて、ランダムに割り当てる場合の操作時間

埋まっている割合（α）に応じた操作時間

挿入時間、非存在要素の検索時間 $\frac{1}{1-\alpha}$

削除、存在要素の検索時間 $-\frac{1}{\alpha}\log_e(1-\alpha)$

ランダムな再ハッシュ法

1次の再ハッシュ、その拡張

ランダムな順列による方法

$h_i(x)=(h(x)+d_i) \bmod B$

シフトレジスタによるd_iの計算

ハッシュ表の再構成

クローズドで N>=0.9B

オープンで N>=2B

ハッシュによる写像の実現

優先度つき待ち行列
(priority queue)

insertとdeletemin を実現するADT

(各要素が優先度を持つ待ち行列)

病院、プロセス割り当て

優先度つき待ち行列の実現

リスト・ソートなし = deltemin が O(n)

リスト・ソート済 = insert が O(n)

半順序付きの2分木 = 両方とも O(log(n))

優先度つき待ち行列の実現

半順序付きの2分木 = 両方とも O(log(n))

deltemin … 一番最後のものを根へ移す

insert … 一番最後に付け加える

ヒープによる実現

複雑な集合構造

多対多の関係 (学生と科目)

配列の場合

多重リスト構造 (複数ポインタのセル)

2重のデータ構造の利用 (名前と順位の例)