

2005年12月06日

2005年度後期・数理解析・計算機数学 プログラミング課題

評価方法について

今回のプログラミング課題と後に配るレポート課題を元にして行う。プログラムにはコメントを含めなければならない。

レポートの提出方法

プログラムソースコードは電子メールの添付ファイルとして提出すること。

提出期限

提出期限は2006年1月10日(火)。

課題

1. 住所録から住所を検索するためのプログラムを作る。追加がたまにしかないと仮定して、内部データを検索しやすいようにするが、そのために追加のコストが高くてもいい。

内部データの型の略称を `db` とする。以下の関数を定義しなさい。

```
val create_db : (string * string list) list -> db
val lookup : string -> db -> string list
```

`create_db` は名前といくつかの文字列からなる住所情報のリストをもらい、検索のために最適化されたデータを返す。`lookup` は名前から住所情報を検索する。名前の数が n だとすれば、`lookup` の計算量は $O(\log n)$ 以内でなければならない(二分探索を使いなさい)。答えがない場合、`Not_found` を起こす。

2. 文字列に対し、以下の二つの関数ができる。

```
val String.length : string -> int
val String.sub : string -> int -> int -> string
```

`String.length s` は `s` の長さを返す。`String.sub s pos len` は `s` の位置 `pos`(先頭は0) から長さ `len` の文字列を取り出す。

この二つの関数を使い、まず関数 `prefix` を定義しなさい。

```
val prefix : string -> string -> bool
```

`prefix s1 s2` は以下の条件が満たされるときだけに `true` になる。

$s2 = s1 \wedge s3$ となるような $s3$ が存在する

\wedge は文字列の結合を表わす．それ以外の場合では `false` になる．

次に, `prefix` を使い, 名前の先頭が `p` である住所情報を全部取り出す関数 `find_all p db` を定義しなさい．

```
val find_all : string -> db -> (string * string list) list
```

答の数が k ならば, `find_all p db` の計算量は $O(k + \log n)$ 以内でなければならない．

注意: 日本語は文字コードによって結合が正しく行われない場合があるので, 名前はローマ字にした方がいい．(EUC や Unicode を使えば日本語でも大丈夫のはず)

3. 既に入力された名前に対して, 住所情報を変更する関数 `update` を定義しなさい．

```
val update : string -> string list -> db -> db
```

`update` の計算量も $O(\log n)$ 以内でなければならない．