

必修課題 2 1冊の本の中に複数回出現する最長部分文字列を探せ (8/8 締切).

本は, Project Gutenberg (<http://promo.net/pg/>) から選ぶこと.

必修課題 1 — 再提出 循環リスト・共有リストの表現法の課題で, その入出力アルゴリズムを記述していない人は, 再提出のこと (6/12 10:30 締切).

随意課題 1 配列の要素に1回目のアクセスで初期値を入れる方法を考えよ.

意図: 余分にメモリを使っても良いが, 全配列の初期化は避けたい.

随意課題 2 自分が選んだアルゴリズムの可視化を行え.

I. 2-3-4 木での挿入 (「A S E A R C H I N G E X A M P L E」)

- (a) 「A S E」 と挿入され, 次に 「A」
が挿入されると, *splitting* が生じる.

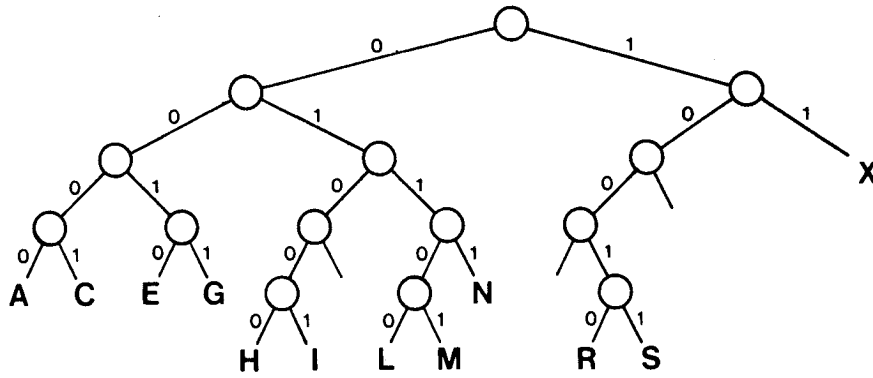
- (b) 「R C H」 と挿入され, 2-node の
子は共に 4-node となるので, 次に
「I」 が挿入されると, 右側の 4-node
が 2つの 2-node に split される.
(親が 3-node になることに注意)

- (c) 「N」 は, 3-node に挿入される.
次に 「G」 が挿入されると, 真中の
4-node が 2個の 2-node に split
される.

- (d) 「E」 を挿入するためには, 親の split-
ting が生じる.

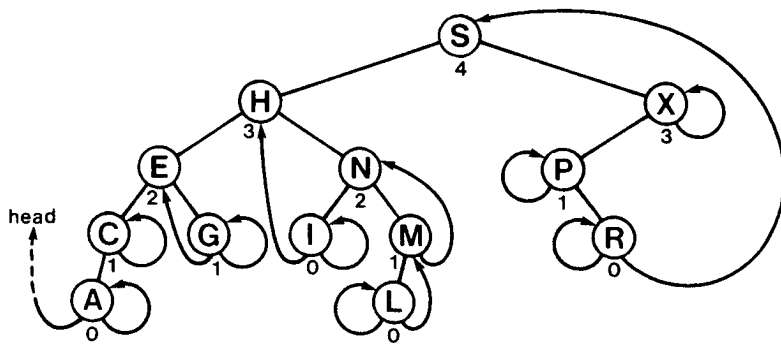
- (e) 「E X A M P L E」 を挿入し終る
と, 次の 2-3-4 木を得る. (同じ要
素は右側に入る)

II. 「A S E A R C H I N G E X A M P L E」 に対するトライ

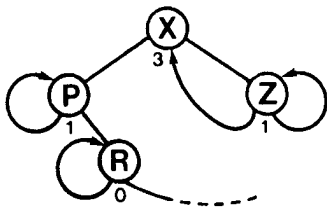


A	00001
C	00011
E	00101
G	00111
H	01000
I	01001
L	01100
M	01101
N	01110
P	10000
R	10010
S	10011
X	11000

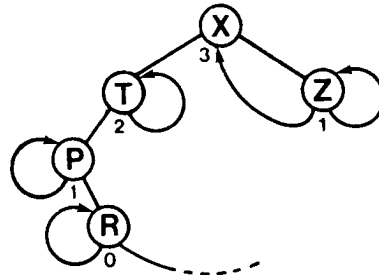
III. 「A S E A R C H I N G E X A M P L E」 に対するパトリシア木



IV. 「Z (11010)」 の挿入



V. 「T (10100)」 の挿入



VI. Patricia 木の探索と挿入プログラム

```

type link=↑node;
    node=record key, info, b: integer; l, r: link end;
var head: link;
function patriciasearch(v: integer; x: link): link;
var f: link;
begin
repeat
f:=x;
if bits(v, x↑.b, 1)=0 then x:=x↑.l else x:=x↑.r;
until f↑.b<=x↑.b;
patriciasearch:=x
end;

```

```

function patriciainsert(v: integer; x: link): link;
var t, f: link; i: integer;
begin
t:=patriciasearch(v, x);
i:=maxb;
while bits(v, i, 1)=bits(t↑.key, i, 1) do i:=i-1;
repeat
f:=x;
if bits(v, x↑.b, 1)=0 then x:=x↑.l else x:=x↑.r;
until (x↑.b<=i) or (f↑.b<=x↑.b);
new(t); t↑.key:=v; t↑.b:=i;
if bits(v, t↑.b, 1)=0
then begin t↑.l:=t; t↑.r:=x end
else begin t↑.r:=t; t↑.l:=x end;
if bits(v, f↑.b, 1)=0 then f↑.l:=t else f↑.r:=t;
patriciainsert:=t
end;

```