

令和2年度

名古屋大学大学院情報学研究科
知能システム学専攻
入学試験問題（専門）

令和元年8月7日

注意事項

1. 試験開始の合図があるまでは、この問題冊子を開いてはならない。
2. 試験終了まで退出できない。
3. 外国人の留学生に限り、日本語と日本語以外の1言語間の辞書1冊に限り使用してよい。電子辞書の持ち込みは認めない。
4. 日本語または英語で解答すること。
5. 問題冊子、解答用紙3枚、草稿用紙3枚が配布されていることを確認すること。
6. 問題は解析・線形代数、確率・統計、プログラミングの3科目がある。これらの全てについて解答すること。なお、選択した科目名を解答用紙の指定欄に記入すること。
7. 全ての解答用紙の所定の欄に受験番号を必ず記入すること。解答用紙に受験者の氏名を記入してはならない。
8. 解答用紙に書ききれない場合は、裏面を使用してもよい。ただし、裏面を使用した場合は、その旨、解答用紙表面右下に明記すること。
9. 解答用紙は試験終了後に3枚とも提出すること。
10. 問題冊子、草稿用紙は試験終了後に持ち帰ること。

プログラミング

一題 2h

26ト一題

プログラム A は、ハッシュ表を用いて正整数の集合を操作するための C 言語プログラムである。ハッシュ値ごとの正整数のリスト構造を、3つの配列 element, next, hfirst を用いて実現している。element 配列は、正整数の集合を格納するための配列である。next 配列は、リストの次の要素が格納されている element 配列の添え字を格納するための配列である。hfirst 配列はハッシュ表のための配列であり、hfirst[h] はハッシュ値 h を持つ正整数のリストの先頭の要素が格納されている element 配列の添え字を表す。ここで、hfirst[h] が -1 のときは、ハッシュ値 h を持つ正整数のリストが空であることを表す。

hashfunc 関数は、ハッシュ関数を表す。search 関数、insert 関数、delete 関数は、それぞれ element 配列に対する正整数の探索、挿入、削除を行う関数である。initarrays 関数は、hfirst 配列、element 配列、next 配列をそれぞれ初期化する関数である。outputarrays 関数は、hfirst 配列、element 配列、next 配列をそれぞれ標準出力に出力する関数である。

プログラム A について、以下の問いに答えよ。

- (1) プログラム A を実行した際に、74 行目の outputarrays 関数の呼び出しにより標準出力に出力される文字列を書け。
1, 2, 3, 5, 7
- (2) プログラム A の 65 行目の int data[] = {1, 2, 3, 5, 7}; を int data[] = {1, 2, 18, 19, 20}; に置き換えて実行した際に、74 行目の outputarrays 関数の呼び出しにより標準出力に出力される文字列を書け。
- (3) data 配列に格納される正整数の集合がどのような特徴を持つときに、search 関数の実行時間が集合のサイズに対して長くなるか答えよ。
↑ 特徴同じ
↑ 値特徴、時間長
- (4) (ア) と (イ) を適切な式で埋めて、delete 関数を完成させよ。
- (5) プログラム A を実行した際に、78 行目の outputarrays 関数の呼び出しにより標準出力に出力される文字列を書け。ただし、プログラム A は (2) の置き換えを行っていないものとする。

Translation of technical terms:

プログラム	program	関数	function
ハッシュ表	hash table	ハッシュ関数	hash function
正整数	positive integer	探索	search
集合	set	挿入	insertion
C 言語	C programming language	削除	deletion
ハッシュ値	hash value	初期化	initialization
リスト	list	標準出力	standard output
構造	structure	出力	output
配列	array	実行	execution
格納する	store	呼び出し	call
要素	element	文字列	character string
添え字	index	実行時間	execution time
空	empty	式	expression

(1) delete → output
↑ 変換 code → data

関数	function
ハッシュ関数	hash function
探索	search
挿入	insertion
削除	deletion
初期化	initialization
標準出力	standard output
出力	output
実行	execution
呼び出し	call
文字列	character string
実行時間	execution time
式	expression

問題の順序実現：
プログラム A

1 #include <stdio.h>
2 #define MAXSIZE
3 #define P
4 #define SENTINEL
5 #define NOTFOUND
6 int hfirst[P];
7 int element[MAXSIZE];
8 int next[MAXSIZE];
9 int avail=-1;
10 int maxnode=0;
11 int hashfunc(int data){
12 return data%P;
13 } 代码分析部分
14 int search(int h, int data)
15 {
16 int pred=-1;
17 if (hfirst[h]==-1) return NOTFOUND;
18 if (element[hfirst[h]]==data) pred=hfirst[h];
19 while (next[pred]!=SENTINEL){
20 if (element[next[pred]]==data) pred=next[pred];
21 }
22 return pred;
23 }
24 }
25 void insert(int h, int data){
26 int u;
27 if (avail==-1){
28 u=avail;
29 avail=next[avail];
30 } else {
31 u=maxnode;
32 maxnode=maxnode+1;
33 }
34 element[u]=data;
35 next[u]=hfirst[h];
36 hfirst[h]=u;
37 } 插入部分
38 void delete(int h, int pred){
39 int u;
40 if (pred!=-1){
41 u=next[pred];
42 next[pred]=(ア);
43 } else {
44 u=hfirst[h];
 hfirst[h]=(イ);
 }
} 删除部分
next[u]=avail
next[3]=



hfirst [0 ~ 67 → element[]

element 装元素 maxnode

next ? pred

第一个 1, 2, 3

```

45     hfirst[h] = (1);
46 }
47 next[u] = avail; avail = u;
48 }
49 void initarrays(){
50     int i;
51     for (i=0; i<P; i++) {hfirst[i]=-1;}
52     for (i=0; i<MAXSIZE; i++) {element[i]=0;}
53     for (i=0; i<MAXSIZE; i++) {next[i]=SENTINEL;}
54 }
55 void outputarrays(int maxnode){
56     int i;
57     for (i=0; i<P; i++) {printf("%d,", hfirst[i]);}
58     printf("\n");
59     for (i=0; i<maxnode; i++) {printf("%d,", element[i]);}
60     printf("\n");
61     for (i=0; i<maxnode; i++) {printf("%d,", next[i]);}
62     printf("\n");
63 }
64 int main(){
65     int data[] = {1,2,3,5,7};
66     int h;
67     int i;
68     int pred;
69     initarrays();
70     for (i=0; i<5; i++){
71         h=hashfunc(data[i]);
72         if (search(h,data[i])==NOTFOUND) insert(h,data[i]);
73     }
74     outputarrays(maxnode);
75     h=hashfunc(1); int code h=1
76     pred=search(h,1); search返回值 pred=-1
77     if (pred!=NOTFOUND) delete(h,pred);
78     outputarrays(maxnode);
79     return 0;
80 }
```

next 下一个
pred → 确定

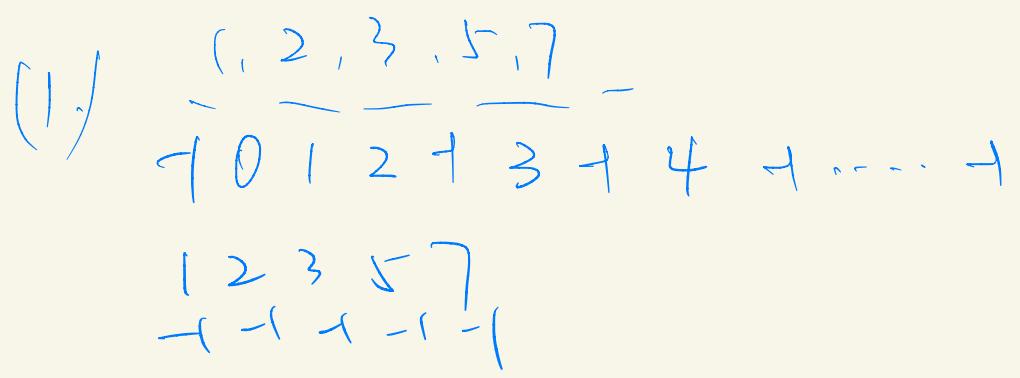
hfirst

-[0] 1 2 + 3 + 4 + - - - +

- |

element [2 3 5 7]

next [-1 -1 -1 -1] 3/3



ht: first hashcode
链表

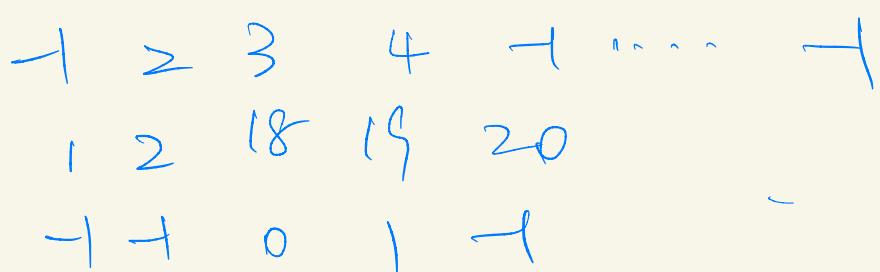
next 下一个
下一个

element 实际值

pre 前驱

avail 空缺地址上

(2) 1, 2, 18, 19, 20



next 下一个

pre 上一个

(3) オーバーヘッドの倍数

(4) $\begin{cases} \text{next} & \text{if } \text{next} \neq \text{null} \\ \text{1} & \text{next} = \text{null} \end{cases}$

(5) -1 + 1 2 + 3 + 4 + ... -1

