

# プログラミング言語実験

## 第3回

### コンピュータ大貧民

— クライアントの基本機能の実装（ペア出し機能の実装） —

2019 年 04 月 22 日、23 日

## 2 場が空の時のペアの提出

### 2.1 ペア提出のアイデア

今のプログラムでは、ペアや階段を出すことができません。そこで、まずは場が空の時に、ジョーカーを考えずにペアを出せるようにしてみましょう<sup>1</sup>。

今回は、次の手順を踏むことで、手持ちのカードからペアを発見し、提出することを考えます。

1. ペアとして出せるカードの情報を持つ配列 `info_table[8][15]` を新しく作成する。
2. `info_table[8][15]` を探索し、一番弱いペアを発見する。
3. 一番弱いペアの情報を提出手として配列に入れる。

新しく作成する `info_table` について説明します。たとえば、手持ちのカードの配列<sup>2</sup>が次の図の通りとします。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0				1			1		1						
1		1		1			1								
2							1								
my_cards :	3														
	4														
	5														
	6														
	7														

このとき、`info_table` として次のような配列を作成します。

<sup>1</sup>この講義で使用する教育用クライアントには実装されていませんが、UECda の標準クライアントには、すでに関数が準備されています。

<sup>2</sup>`my_cards` や `own_cards` という名前で、プログラム内には登場します。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0															
1															
2															
info_table :	3														
	4	1		2			3		1						
	5														
	6														
	7														

何をしているかというと、info\_table の 4 行目に、その強さのカードが手持ちに何枚あるかを書いているだけです。作成した後、ある枚数以上のペアや最大サイズのペアを発見したいなら、それらの組み合わせがあるかを 4 行目を見ることによって発見します。どの強さのカードをペアとして出すかが決まったら、今度は手持ちのカードの配列のその列を、提出する配列 select\_cards にコピーすれば良いことになります。

次の小節から、この処理を行うプログラムを実際書いていきます。

## 2.2 info\_table の作成

手札情報 my\_cards から info\_table を生成する関数 make\_info\_table を、daihinmin.c に作成します。引数は info\_table と my\_cards です。ですので、プロトタイプ宣言は次のようになります。

Listing 1: make\_info\_table のプロトタイプ宣言

```
1 void make_info_table(int info_table[8][15], int my_cards[8][15]);
```

作業：プロトタイプ宣言の追加

make\_info\_table のプロトタイプ宣言を daihinmin.h に新たに追加しましょう。

続いて、実際に make\_info\_table が行う処理について考えましょう。info\_table[4][i] に入る値は、その値のカードの枚数です。つまり、my\_cards[0][i], my\_cards[1][i], my\_cards[2][i], my\_cards[3][i] の和です。式で書くと、info\_table[4][i]=my\_cards[0][i]+my\_cards[1][i]+my\_cards[2][i]+my\_cards[3][i] となります。この計算を、i=1~13 に対して行うプログラムを書けばよいことになります。

作業：実装内容の検討

make\_info\_table の実装内容（実際のコード）を考えてみましょう。

作業：関数 make\_info\_table の追加

make\_info\_table を daihinmin.c に新たに追加しましょう。

作業：make\_info\_table の説明

make\_info\_table が、どのようにして必要な処理を行っているか、説明しなさい。

## 2.3 ペアの発見と提出カードの設定

info\_table ができたので、ここからペアを発見し、提出するカードとして設定します。ペアとして出せる手を発見するためには、info\_table の 4 行目を調べていき、2 以上の値が入っているところを探せばよいことになります。

今回は、一番弱いペアを発見する関数 search\_low\_pair を作成します。この関数は、自分の手札と info\_table から、一番弱いペアを発見し、その結果を dst\_cards に設定します。加えて、ペアを発見できたときは、返り値として 1 を、見つからなかった場合は 0 を返します。たとえば、

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0				1			1		1						
1		1		1			1								
2							1								
3															
4															

と

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0															
1															
2															
3															
4		1		2			3		1						

から、

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0				1											
1				1											
2															
3															
4															

を設定します（5 行目以降は本当はありますが、省略しています）。プロトタイプ宣言は次のようになります。

Listing 2: search\_low\_pair のプロトタイプ宣言

```
1 int search_low_pair(int dst_cards[8][15], int info_table[8][15],
2                     int my_cards[8][15]);
```

作業：search\_low\_pair のプロトタイプ宣言の追加

search\_low\_pair を daihinmin.h に新たに追加しましょう。

この処理を実現するためには、どうしたらよいでしょうか。今回は、次のような方針で行くことにします。まず、info\_table の 4 行目を左から右に順番に見ていき、2 以上の値を探します。そして、見つかった列を my\_cards から dst\_cards にコピーします。

作業：実装内容の検討

`search_low_pair` の実装内容を考えてみましょう。

作業：`search_low_pair` の追加

`search_low_pair` を `daihinmin.c` に新たに追加しましょう。

作業：`search_low_pair` の説明

`search_low_pair` が、どのようにして必要な処理を行っているか、説明しなさい。

## 2.4 提出するカードとして設定する

`info_table` を作成し、そこから最も弱いペア情報を取り出すことまでできるようになりました。ただ関数を準備しただけですので、まだカードを出すことはできません。最後に、`select_cards_free` で、作成した関数を呼び出すことにより、場が空のときは最も弱いペアを優先的に出すようにしましょう。そのために、今回は、次のことを `select_cards_free` 内で行うように追記します。

1. ペア情報を保存する配列 `info_table` を宣言する。
2. 自分のカード情報 `my_cards` からペア情報を作成し、配列 `info_table` に保存する。
3. 自分のカード情報 `my_cards` とペア情報 `info_table` から、最弱のペアを発見し、`selesct_cards` に保存する。
4. ペアが無かった場合は、最も弱いカードを単騎で出すように、`selesct_cards` に情報を保存する。

作業：実装内容の検討

`select_cards_free` の実装内容を考えてみましょう。

作業：`select_cards_free` の修正

`select_cards.c` の `select_cards_free` を実装しましょう。

作業：`select_cards_free` の説明

`select_cards_free` が、どのようにして必要な処理を行っているか、説明しなさい。

これで、場が空のときは、最弱のペアを優先的に提出するようになりました。`info_table` には、すべてのペアの候補が保存されていますので、この配列の探索方法を変更すれば、他のペアを優先することもできます。

余裕のある人向け：最大枚数ペアの提出

場が空の時、手持ちのカードから、最大枚数のペアを優先的に提出したい。どのようにすれば実現できるか考えなさい。