

アルゴリズムとデータ構造 プログラミング演習

森 立平 mori@c.titech.ac.jp

2016 年 7 月 1 日

今日の目標

- 動的計画法（「宝詰め込み問題」、最適解の探索も含む）のプログラムを書けるようになる.
- 「レポート」に必要なプログラムを書く.

今日の課題はなし

今日のワークフロー

1. まだ「宝詰め込み問題」の補助問題の定義と補助問題の解が満たす再帰的な関係式（漸化式）が導出できていない場合はそれらをもとめる.
2. プログラムの雛形を保存する.
3. プログラムの雛形をもとに「宝詰め込み問題」のプログラムを書く（いきなりボトムアップ法で書いてもよいし、まずはトップダウン法のプログラムを確実に書くのもよい）. まずは最適値を正しく計算するプログラムを書くことを目標にするとよい.

1 宝詰め込み問題とプログラムの仕様について

宝詰め込み問題を改めて記述する.

入力: $n, W, C[0], \dots, C[n-1], V[0], \dots, V[n-1]$.

出力: $S \subseteq \{0, 1, \dots, n-1\}$ で $\sum_{i \in S} C[i] \leq W$ を満たすものの中で $\sum_{i \in S} V[i]$ を最大化するもの. および, そのような S に対する $\sum_{i \in S} V[i]$ の値.

次にレポートで提出してもらおう宝詰め込み問題のプログラムの仕様を記述する. 宝詰め込み問題のプログラムは標準入力から問題の入力を受け取って, 標準出力に問題の解を出力することにする (標準入力, 標準出力の説明は次章でおこなう). 入力のフォーマットは

```
n
W
C[0] C[1] ... C[n-1]
V[0] V[1] ... V[n-1]
```

とする. ここで, 改行とスペースは同じように扱うとする. 例えば

```
5
13
1 2 5 7 8
1 3 4 5 10
```

は入力の一例である. 次に出力のフォーマットは

最適値
最適解 (宝のインデックスを昇順で並べる)

とする.

```
14
0 1 4
```

一般的に最適解は一意には定まらないが, 何かひとつ最適解を出力すればよい.

2 標準入出力について

プログラムに対して入力する方法，出力する方法は様々である．その中でも標準入出力を使うのは最も基本的で柔軟な方法である．ここは標準入力から「宝詰め込み問題」の入力を受け取るプログラムを雛形として与える．

宝詰め込み問題のプログラムの雛形

```
#include <stdio.h>

#define N 1000
#define WMAX 100000

unsigned int C[N+1];
unsigned int V[N+1];

int main(){
    int i;
    unsigned int n, W;

    if(scanf("%u", &n) != 1) return 1;
    if(scanf("%u", &W) != 1) return 1;
    if(n > N){
        puts("Too many items");
        return 1;
    }
    if(W > WMAX){
        puts("Too large bag");
        return 1;
    }
    for(i = 0; i < n; i++){
        if(scanf("%u", C+i) != 1) {
            puts("Error");
            return 1;
        }
    }
    for(i = 0; i < n; i++){
        if(scanf("%u", V+i) != 1) {
            puts("Error");
            return 1;
        }
    }
    /*
    この時点で宝の数は n に，袋の限界容量は W に，
    宝の重さは配列 c に，宝の価値は配列 v に
    代入されているとしてよい
    */

    /* 宝の重さと価値を出力（最終的には次の3行は消すこと）*/
    for(i = 0; i < n; i++){
        printf("%d, %d\n", C[i], V[i]);
    }

    return 0;
}
```

この雛形を用いて，既に問題の入力は C 言語プログラムの変数に代入されているとしてプログラムを書いてよい．また標準出力へ出力するためには，今までと同じように printf を使えばよいので，特に何も気にする必要はない．

最後に実際にプログラムに入力を与え出力を受け取る方法を説明する。まずテキストファイル `input.txt` にプログラムへの入力を書いておく。例えば

```
5
13
1 2 5 7 8
1 3 4 5 10
```

と `input.txt` に書いておく。それから

```
./a.out < input.txt
```

をターミナルから実行すれば、`input.txt` に書いてある入力をプログラムで受け取ることができる。プログラムの正しさを確認するための入力を `example_input.txt` に用意した。プログラムの出力が `example_output.txt` と一致すればよい（最適解は唯一である）。