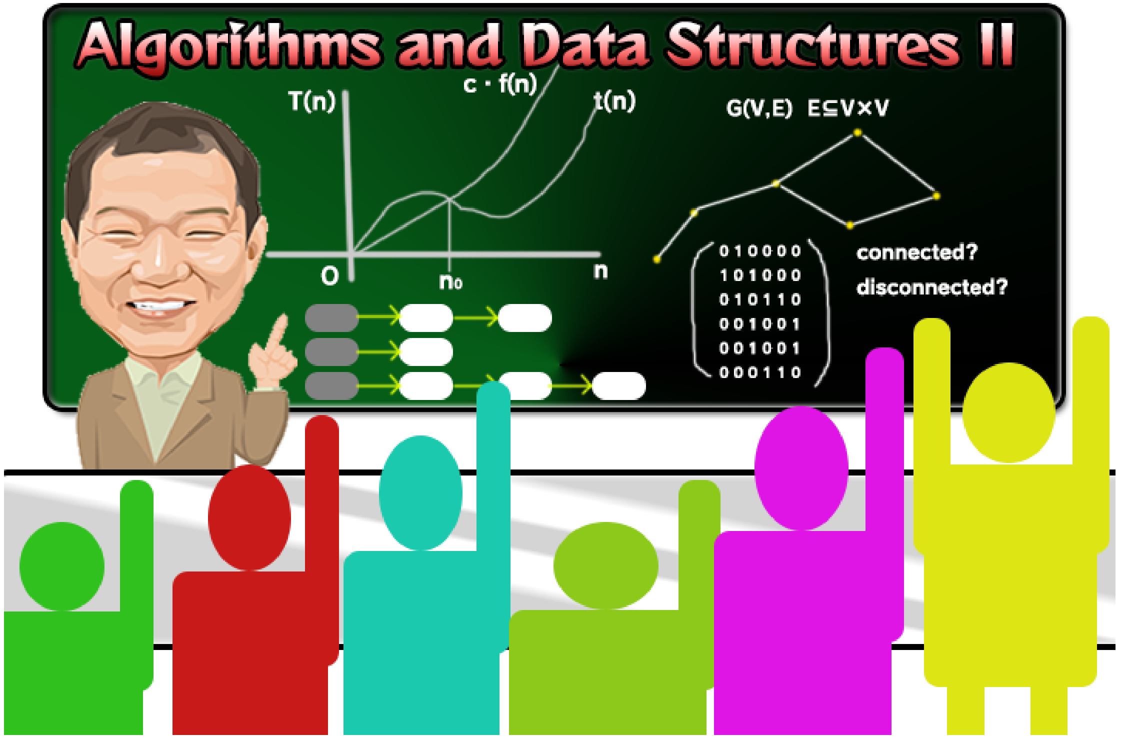

アルゴリズムとデータ構造 II
Algorithms and Data Structures II

Exercise 10. Algorithm Design Techniques III

Exercise 08-11のために

アルゴリズム設計のためのストラテジー
（問題を解くための、広い意味での解法の考え方）

- Greedy Algorithm (Ex08)
- Divide and Conquer (Ex09)
- **Dynamic Programming (Ex10)**
- Backtracking (Ex11)

[Ex10] Dynamic Programming (DP/動的計画法)

ある目的の計算のために、より小さい問題にしてから順に解く

⇒ 小さい問題を解いてからだんだん大きく
⇒ 大きい問題を解くときに、小さい問題で得た解を再利用

・小さい問題で得た解を次々と記録するためのテーブルが必要
（Dynamic Programming の Programming とはこのテーブルを作る操作）

・漸化式の形で表される問題もDP

☆力技では実用的ではない、高い計算量になるプログラムも
DPを使うことで効率のよいプログラムに書き換えられる

Dynamic Programming

動的計画法の基本的な流れ

・求めるべきものを、それを得るまでに必要な部分解を含めて全てを格納するテーブル（配列）を準備する

・テーブルを適切に初期化する

・ある値を求めるために、既に計算した結果を使って更新ができるようにする
⇒既に計算した結果についてはテーブルを参照して取り出す
(ここが漸化式のようなイメージ)

※いかにテーブルを作るかは問題により異なるので、練習するしかない

Dynamic Programming

【問題１】の類題（問題１の手引き）

１円玉、７円玉、12円玉　が与えられていて　30円をちょうど支払う

⇒　できるだけ最小の硬貨枚数で30円になる組み合わせの数を求める

・１円を払う方法から順番に考えていって、30円まで拡大する

最低でも、0円から30円までを保持する配列（テーブル）が必要。

・初期条件（結果が自明なものは、とりあえずテーブルを埋めておく）

0円　⇒　硬貨0枚

1円　⇒　硬貨1枚（1円玉）

7円　⇒　硬貨1枚（7円玉）

12円　⇒　硬貨1枚（12円玉）

Dynamic Programming

1円の支払い方は既に確定済み。

⇒その状態に、各種類の硬貨を1枚追加した場合の支払いを考える

⇒今の場合、2円、8円、13円の支払い方法を検討可能

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
[Initial step]:
  0  1 -- -- -- -- --  1 -- -- -- --  1 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[1 step (1 coin)]:
  0  1  2 -- -- -- --  1 -- -- -- --  1 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[1 step (7 coin)]:
  0  1  2 -- -- -- --  1  2 -- -- --  1 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[1 step (12 coin)]:
  0  1  2 -- -- -- --  1  2 -- -- --  1  2 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[2 step (1 coin)]:
  0  1  2  3 -- -- --  1  2 -- -- --  1  2 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[2 step (7 coin)]:
  0  1  2  3 -- -- --  1  2  3 -- --  1  2 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[2 step (12 coin)]:
  0  1  2  3 -- -- --  1  2  3 -- --  1  2  3 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[3 step (1 coin)]:
  0  1  2  3  4 -- --  1  2  3 -- --  1  2  3 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[3 step (7 coin)]:
  0  1  2  3  4 -- --  1  2  3  4 --  1  2  3 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[3 step (12 coin)]:
  0  1  2  3  4 -- --  1  2  3  4 --  1  2  3  4 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

Dynamic Programming

次に、1円増やして、2円から各種類の硬貨を1枚追加した場合の支払いを考える　⇒今の場合、3円、9円、14円の支払い方法を検討可能

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
[Initial step]:
 0  1 -- -- -- -- --  1 -- -- -- --  1 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[1 step (1 coin)]:
 0  1  2 -- -- -- --  1 -- -- -- --  1 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[1 step (7 coin)]:
 0  1  2 -- -- -- --  1  2 -- -- --  1 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[1 step (12 coin)]:
 0  1  2 -- -- -- --  1  2 -- -- --  1  2 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[2 step (1 coin)]:
 0  1  2  3 -- -- --  1  2 -- -- --  1  2 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[2 step (7 coin)]:
 0  1  2  3 -- -- --  1  2  3 -- --  1  2 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[2 step (12 coin)]:
 0  1  2  3 -- -- --  1  2  3 -- --  1  2  3 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[3 step (1 coin)]:
 0  1  2  3  4 -- --  1  2  3 -- --  1  2  3 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[3 step (7 coin)]:
 0  1  2  3  4 -- --  1  2  3  4 --  1  2  3 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[3 step (12 coin)]:
 0  1  2  3  4 -- --  1  2  3  4 --  1  2  3  4 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

Dynamic Programming

同様に、1円増やして、3円から各種類の硬貨を1枚追加した場合の支払いを考える　⇒今の場合、4円、10円、15円の支払い方法を検討可能

この段階では、まだ　<u>暫定的な最小使用硬貨数</u>　であることに注意

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
[Initial step]:
 0  1 -- -- -- -- --  1 -- -- -- --  1 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[1 step (1 coin)]:
 0  1  2 -- -- -- --  1 -- -- -- --  1 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[1 step (7 coin)]:
 0  1  2 -- -- -- --  1  2 -- -- --  1 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[1 step (12 coin)]:
 0  1  2 -- -- -- --  1  2 -- -- --  1  2 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[2 step (1 coin)]:
 0  1  2  3 -- -- --  1  2 -- -- --  1  2 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[2 step (7 coin)]:
 0  1  2  3 -- -- --  1  2  3 -- --  1  2 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[2 step (12 coin)]:
 0  1  2  3 -- -- --  1  2  3 -- --  1  2  3 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[3 step (1 coin)]:
 0  1  2  3  4 -- --  1  2  3 -- --  1  2  3 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[3 step (7 coin)]:
 0  1  2  3  4 -- --  1  2  3  4 --  1  2  3 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
[3 step (12 coin)]:
 0  1  2  3  4 -- --  1  2  3  4 --  1  2  3  4 -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

Dynamic Programming

・11円に硬貨を追加するケースまでワープ（飛ばした部分も同じ方針）
11円から各種類の硬貨を1枚追加した場合の支払いを考える
⇒今の場合、12円、18円、23円の支払い方法を検討可能

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
[11 step (1 coin)]:	0	1	2	3	4	5	6	1	2	3	4	5	1	2	2	3	4	5	7	2	3	4	5	--	--	--	--	--	--	--	--
[11 step (7 coin)]:	0	1	2	3	4	5	6	1	2	3	4	5	1	2	2	3	4	5	6	2	3	4	5	--	--	--	--	--	--	--	--
[11 step (12 coin)]:	0	1	2	3	4	5	6	1	2	3	4	5	1	2	2	3	4	5	6	2	3	4	5	6	--	--	--	--	--	--	--

18円の支払い方は、最小「７」枚だったのが、「６」枚に変わっている
・11円の段階で５枚消費済み
⇒ １円を+１枚（６枚で１２円払える ⇒ 12円玉１枚の方が良い）
⇒ ７円を+１枚（６枚で１８円払える ⇒ 12円玉+1円玉×6より良い）
⇒ 12円を+１枚（６枚で２３円払える ⇒ 23円の支払い方法は未決定）

Dynamic Programming

・まとめると…

金額が a のとき、 dp[i] (0≦ i ≦ a)には、金額 i を支払うための
最小の硬貨の枚数（dp[i]枚）が入っている

硬貨の額面が b のものが n 種類あるとすると
１から順に a の支払い方法を考える
b[1] を+１枚する： dp[i] +1 枚と dp[i+ b[1]] 枚のどっちが最小？
b[2] を+１枚 する： dp[i] +1 枚と dp[i+ b[1]] 枚のどっちが最小？
……
⇒ b[n] を+１枚 する： dp[i] +1 枚と dp[i+ b[n]] 枚のどっちが最小？
・i 円払うときの最小枚数に硬貨b[n]を１枚を追加したときと
・以前に決定した金額 i+b[n] を支払う暫定枚数を使うときと
比べればどちらが最小枚数になる？

Dynamic Programming

30ステップの処理を行ったのち、dp[30]を見れば
30円のときの硬貨の合計枚数がわかる！

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
[30 step (1 coin)]:
 0  1  2  3  4  5  6  1  2  3  4  5  1  2  2  3  4  5  6  2  3  3  4  5  2  3  3  4  4  5  6
[30 step (7 coin)]:
 0  1  2  3  4  5  6  1  2  3  4  5  1  2  2  3  4  5  6  2  3  3  4  5  2  3  3  4  4  5  6
[30 step (12 coin)]:
 0  1  2  3  4  5  6  1  2  3  4  5  1  2  2  3  4  5  6  2  3  3  4  5  2  3  3  4  4  5  6
Totally, you used 6 coins for 30 cents.
```

<u>内訳は分からないのか？</u>　⇒ dp[ ]のテーブルをdp[30]から逆にたどる
30円 のとき、6枚。
30-12=18が 「dp[18]が5枚になっていれば、12円玉を使っている」⇒ 6 だから不使用
30-7=23 ⇒ 「５」になっているので７円玉を使っている
23円から同様に、各硬貨の額面を引いて…
23-7=16 ⇒ dp[16]=4だから、7円玉をまた使っている（2枚め）　以下同様に…
16-7 = 9 , 9-7=2, 2-1=1, 1-1=0 より,　7円玉×4 + 1円玉×2 の合計 6枚 で30円