貪欲法
~GREEDY ALGORITHM~

「その場で最善」を選択することを繰り返すアルゴリズム

概要

- このアルゴリズムは問題の要素を複数の部分問題に分割し，それぞれを独立に評価を行い，評価値の高い順に取り込んでいくことで解を得るという方法である．(by Wikipedia)
- 1つのルールに従って，その中で最適な解を選択するアルゴリズム
- このアルゴリズムが存在するならば，シンプルかつ高速に解が求まる．

説明(1)

- 例題の説明
  - 1円玉，5円玉，10円玉，50円玉，100円玉，500円玉が，それぞれC_1，C_5 C_{10}，C_{50}，C_{100}，C_{500}枚ずつあります．できるだけ少ない枚数の硬貨でA円を支払いたいと考えている．何枚の硬貨を出す必要があるでしょうか？なお，そのような支払い方は少なくとも1つは存在するとします．

    <制約>

    $0 <= C_1$，C_5 C_{10}，C_{50}，C_{100}，$C_{500} <= 10^9$

    $0 <= A <= 10^9$

説明(2)

- ここでは,「大きな額から」というのがルールである.
  - for文をうまく使おう！
- 1つのルールしかないので実装が楽
- もちろんJavaでね

説明(3)~Java~

```
import java.util.Scanner;

public class Main {

	public static void main(String[] args) {

		int[] C = {1, 5, 10, 50, 100, 500};

		Scanner sr = new Scanner(System.in);
		System.out.print("plz enter Fee :");
		int A = sr.nextInt();
		int ans = 0;

		for(int i=5; i>=0; i--) {
			int t = A / C[i];
			if(t > 0) {
				System.out.println(C[i] + " : " + t);
			}

			A -= t*C[i];
			ans += t;
		}

		System.out.println("sum :" + ans);
	}

}
```