



動的計画法


Knapsack問題

計算機アルゴリズム特論：2015年度

只木進一


問題の確認

- n 個の品物の集合 $G = \{g_n | 0 \leq n < n\}$
 - 各品物 g_i には、予め重量 $g_{i.w}$ と価値 $g_{i.v}$ が設定されている
- $G' \subseteq G$ とする
 - 制約 $\sum_{g \in G'} g.w \leq W$ の下で $\sum_{g \in G'} g.v$ を最大化する G' を求める



荷物のクラスGood

```
public class Good {  
  
    private final int weight;//重量  
    private final int value;//価値  
    private final String label;//ラベル  
  
    ...  
}
```




ナップザックのクラス

```
public class Knapsack implements Cloneable {  
    private List<Good> goods;  
    private int value;  
    private int weight;  
  
    public void addGood(Good g) ;//荷物を追加  
    public Knapsack clone();//コピーを作成  
}
```

解法の抽象クラス

```
public abstract class AbstractKnapsack {  
    protected final List<Good> goods;//荷物の一覧  
    protected final int maxWeight;//許容重量  
    protected Knapsack knapsack;//ナップザックの最終的状态  
    protected boolean debug = true;  
    protected int count;  
}
```



```
public AbstractKnapsack(List<Good>
goods, int maxWeight); //コンストラクタ
public int doExec() ;//解法の入口
abstract protected Knapsack doRec(int i,
int w;//解法の実装部
}
```