ABC 032 解説

三上和馬(@kyuridenamida)

A - 高橋君と青木君の好きな数

問題概要

- 整数 $a, b, n(1 \leqq a, b \leqq 100, n \leqq 20000)$ が与えられる.
- aとbの公倍数で，n以上で最小のものを求めよ.

- 例) a=2,b=3,n=8 → 12 (n以上で最小の，2と3の公倍数は12)

解法

- 与えられたnを，aとbの両方で割り切れるようになるまで愚直にインクリメントして，割り切れたらそれを出力．

- 高々a*b-1回インクリメントすればそのような数は見つかるので，時間計算量は O(ab) で間に合う．

別解

- まずaとbの最大公約数$\gcd(a,b)$をユークリッドの互除法で求める.
- そしてaとbの最小公倍数$\mathrm{lcm}(a,b) = \frac{ab}{\gcd(a,b)}$を求めてから,

$$\mathrm{lcm}(a,b) * \left(\frac{n}{\mathrm{lcm}(a,b)} \text{を切り上げたもの}\right)$$

- を計算するとそれが答え.

[余談]
一般に$\frac{a}{b}$の切り上げは,
$\frac{a+b-1}{b}$を整数で計算すると求まる

- この方法だとn,a,bが大きくても解けるが今回は必要ない.

B -高橋君とパスワード

問題概要

- 文字列sに含まれる長さkの部分文字列で相異なるものの数を数えよ.
- 1≦|s|,k≦300
- 例) s=“abcabc”,k=2→ 答え3 (∵ {”ab”,”bc”,”ca”}の3通りがありうる)

解法

- 存在する長さkの部分文字列をすべて列挙したリストを作ってから，重複を取り除き，その個数を出力する.
- 重複を取り除く方法はいろいろあるが，多くの言語に存在するset型などの集合型を用いると，簡単に重複を取り除くことができる.

- 計算量は集合型の実装方法に依存する.
  - 平衡二分木による実装の場合 $O(|s|^2 \log |s|)$
  - 平衡二分木による実装の場合 $O(|s|^2)$

  ※文字列比較のオーダーはO($|s|$)なことに注意

- 集合型を用わず，列挙したリストに対して二重ループで愚直に重複除去を行っても，$O(|S|^3)$程度の計算量で解ける.

C – 列

C - 列

- 長さNの非負整数列$S = s_1, s_2, \ldots, s_N$と整数Kが与えられる.
- Sの連続する部分列$s_i, s_{i+1}, \ldots, s_j$のうち，条件

$$\prod_{k=i}^{j} s_k \leq K$$

を満たす部分列で最長のものの長さを求めよ.

- 条件を満たす部分列が存在しない時は0を出力

- $1 \leqq \mathrm{N} \leqq 10^5, 0 \leqq s_i, K \leqq 10^9$

例

- S={4,3,1,1,2,10},K=6　→　答え 4
- S={10,10,10,10,0,10},K=10　→　答え 6
- S={10,10,10,10,10,10},K=9　→　答え 0
- S={1,2,3,4},K=0　→　答え 0

考察

[数列に1つでも0という値が含まれているケース]

- 答えはN

[それ以外のケース]

- ありうる部分列の左端を全通り試すことを考える.
  - その上で右端を全通り試していたら間に合わなさそう…?
  - 右端を少しずつ伸ばし要素の積がKを超えた時点で打ち切れば良い?
  - 全ての要素が2以上だったら右端の候補としてありうるのは$\log_2 K$通りくらい
    - なぜなら1つ要素が増えるごとに2倍以上になっていき，すぐKを超えるから
  - しかし，連続した1があると右端の候補数がO(N)になってしまう.
  - 連続した1を圧縮すれば良さそう！

考察 - 連続した1を圧縮する

- 圧縮例

  S = {2,3,3,1,1,1,1,2,1,1,1} → {2,3,3,(1が4コ),2,(1が3コ)}

- このようにすると，右端を少しずつ伸ばしていく際，連続した1の部分は一気に伸ばすことができる.
- 最悪ケースはS={1,2,1,2,1,2,1…}というふうなケースだが，
  しかしこの場合でも各左端から高々 $2\log_2 K$ コ程度しか候補が無い.

解法

- 数列を圧縮する
- 圧縮後の数列に対し，全左端から要素の積がKを超えない範囲で右端を少しずつ伸ばしていくという方法で全通り試す．
- 時間計算量 O(N log K)

- しかしもっと汎用的で計算量も少ない解法がある

別解① – 尺取法

- この問題は尺取法で解くことができる.
  - 尺取法とは一次元配列に対して，左端と右端の2つのインデックスを持って片方を進めたりして解を求める手法.
- 以下のように尺取法を行うと，”全体で”O(N)の時間計算量で解ける.
  1. 今の左端(初期は1番目)に対して，要素の積がK以下の間，出来るだけ右端を伸ばす.
  2. 今の区間の長さが解より大きければ解を更新
  3. 伸ばせなくなったら1つ左端を右に動かす(=縮める). ただし左端を動かせないなら終了.
  4. 1に戻る
- 左端が右端を追い越さないように注意して実装すること.
- 実装は開区間で持つのが良いと思われる

尺取法のイメージ

- 以下のケースを考える
  - S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

①最初の状態

[左端,右端)=[1番目,1番目) ※開区間です

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

②伸ばせるだけ伸ばす

[左端,右端)=[1番目,3番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

①左を1つ進める

[左端,右端)=[2番目,3番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

②伸ばせるだけ伸ば..せない

[左端,右端)=[2番目,3番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

①左を1つ進める

[左端,右端)=[3番目,3番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

②伸ばせるだけ伸ば..せない

[左端,右端)=[3番目,3番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

①左を1つ進める(※左が右端を追い越すので右端も進める)

[左端,右端)=[4番目,4番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

②伸ばせるだけ伸ばす

[左端,右端)=[4番目,7番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

①左を1つ進める

[左端,右端)=[5番目,7番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

②伸ばせるだけ伸ばす

[左端,右端)=[5番目,8番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

①左を1つ進める

[左端,右端)=[6番目,8番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

②伸ばせるだけ伸ば..せない

[左端,右端)=[6番目,8番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

①左を1つ進める

[左端,右端)=[7番目,8番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

②伸ばせるだけ伸ば..せない

[左端,右端)=[7番目,8番目)

S={1,2,10,3,3,1,2}, K=9

尺取法のイメージ

③結局最長は3だとわかる

S={1,2,10,3,3,1,2}, K=9

別解② – logを取って和の問題に帰着

- 入力の対数を取ると，和の問題に帰着できるので，累積和と二分探索を用いて解ける.
- ただし，誤差に気をつける必要がある.
- 詳細は省略

D –ナップサック問題

問題概要

- 以下のような0/1ナップサック問題を解いてください.
  - N個の荷物があって，それぞれの荷物には価値と重さが割り当てられている.
  - 重さの総和がW以下となるように荷物を選ぶとき，価値を最大化してください.
  - ただし,
    A) Nが30以下のケース
    B) 全ての荷物の価値が1000以下のケース
    C) 全ての荷物の重みが1000以下のケース
  - の少なくともの1つが成り立つケースしかデータセットに存在しない.
- $1 \leqq N \leqq 200$
- $1 \leqq$ 荷物の価値v_i, 荷物の重み$w_i \leqq 10^9$
- $1 \leqq W \leqq 10^9$

はじめに

- ナップサック問題は，動的計画法(DP)を用いる例題としてとても有名
  - 聞いたことがない人は，本やインターネットに図解が溢れかえっているので是非調べてみましょう
  - 今回は文章だけです．

- ナップサック問題は，
  - 重さの和や価値の和に関して上限がないとき

  →NP困難問題として有名で，多項式時間では解けない

  - 重さの和や価値の和に制約があるとき

  →計算量がそれらの値に依存する擬多項式時間アルゴリズムがある
  (よくあるDPのことです)

考察

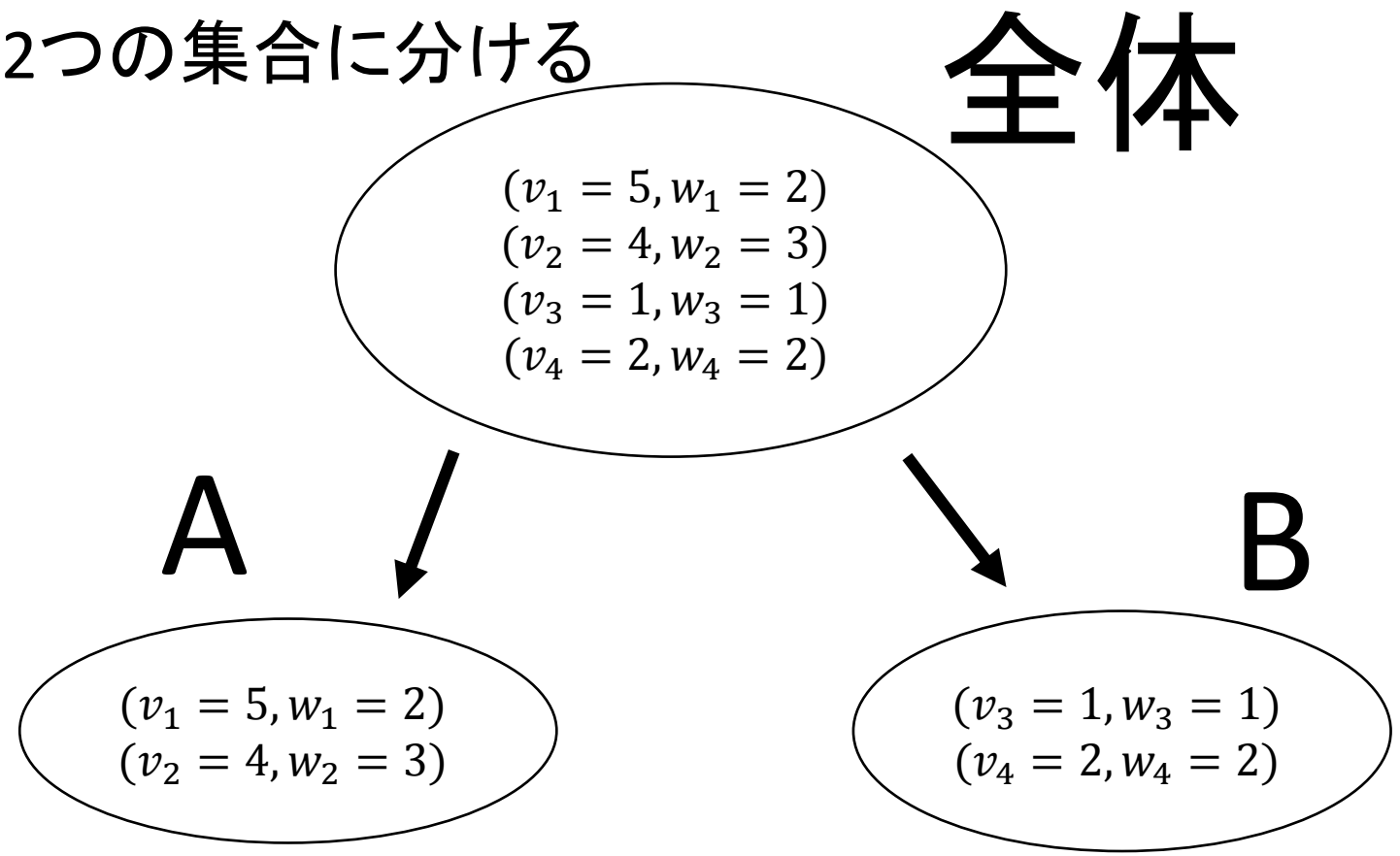
- 各データセット毎に場合分けして解く他ない.
  A) Nが30以下のケース
  → DPは困難∧愚直な全列挙は難しい(枝刈り探索なら通るかも..?)
  B) 全ての荷物の価値が1000以下のケース
  → DPできる
  C)全ての荷物の重みが1000以下のケース
  →DPできる
- それぞれについて考える

考察＆解法　(A) Nが30以下のケース

- 価値・重みに制約が無いため動的計画法は困難
- $O(2^N)$で全列挙は間に合わなさそう.
- 与えられた荷物の集合を半分ずつの集合A,Bに分けて，それぞれの集合で組み合わせを全列挙し，それらの結果をうまくマージすると，$O(2^{\frac{N}{2}} * \log S)$程度の計算量で解ける($S = 2^{\frac{N}{2}}$とする).
  - そのために，まず集合Aに対して，重さの和x以下で達成できる最大価値を$O(\log S)$程度で求めれるようにしておく必要がある.
  - S通りある(重みの和,価値の和)のペアを辞書順ソートして，重みに対して価値が単調増加するようにリストを作っておけば二分探索できる.
  - その後，集合Bのある組み合わせを試して，その組み合わせを使った後余る重みを集合Aに割り振る. この時達成できる最大価値を先ほどのリストに対する二分探索等で高速に求める.

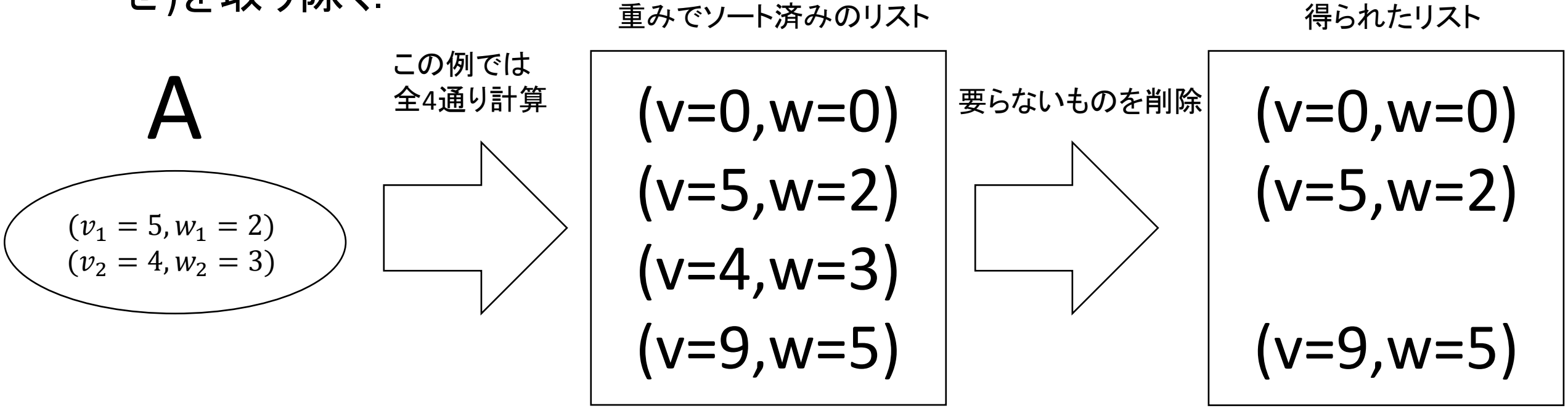
アルゴリズムの流れ (A) Nが30以下のケース

① 荷物を2つの集合に分ける



アルゴリズムの流れ (A) Nが30以下のケース

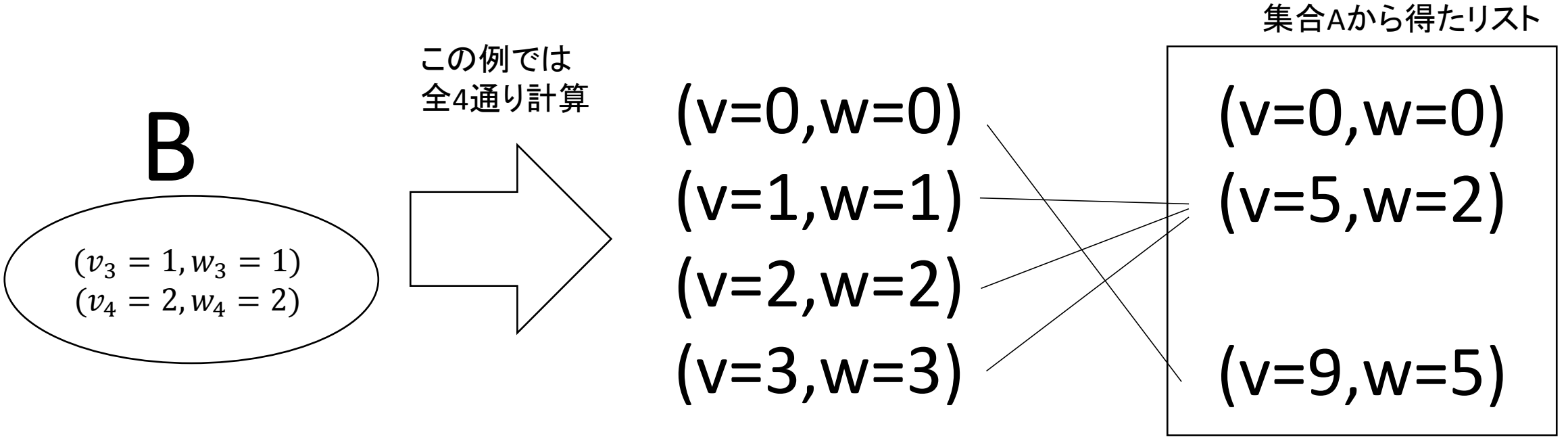
② Aの全組み合わせを計算してリストを作り，明らかに損するもの(それ未満の重みで，それ以上の価値を達成するものがある組み合わせ)を取り除く.



アルゴリズムの流れ (A) Nが30以下のケース

③ Bの全組み合わせに対して，先ほど計算したリストのどの要素とくっつければ良いかを，重みに対する二分探索で計算し，マージする

例)ナップサックのサイズが5の場合を考えると以下のようになる



考察＆解法 (B)荷物の価値が1000以下

- $V_{MAX} = 1000$とおくと，価値の総和は高々 $N * V_{MAX}(= 20万)$ 以下.
- 価値に対して重みを最小化するDPができる.
- $dp[i][j]$:= i番目のアイテム$(i = 1..N)$まで使って，価値の和jを達成する重みの和の最小値

と定義すると，以下の漸化式を計算すれば解ける

- $dp[0][0] = 0, dp[0][1..NV_{MAX}] = \infty$
- $dp[i][j] = \min(dp[i-1][j], dp[i-1][j-v_i] + w_i$(但し$j - v_i \geqq 0$の時))

- この漸化式を$i = 0..N$に対して計算した後，重みの和が W 以下となる最大の価値をテーブルを参照して求めれば良い.

考察＆解法　(B)荷物の価値が1000以下

- 実は，価値の逆順に更新ループを行うと，テーブルのサイズが$O(NV_{MAX})$で済み，添字iについて考慮しなくてよくなる．実装も軽い．
- 時間計算量は $O(N^2V_{MAX})$　空間計算量は $O(NV_{MAX})$

配列$dp[0..N][0..NV_{MAX}]$を用意
$For\ i\ =\ 1..NV_{MAX}$
　$dp[0][i] = \infty$
$dp[0][0]\ =\ 0$
$For\ i\ =\ 1..n$
　$For\ j\ =\ 0\ ..(v_i - 1)$
　　$dp[i][j]\ =\ dp[i-1][j];$
　$For\ j\ =\ v_i\ ..NV_{MAX}$
　　$dp[i][j]\ =\ \min(dp[i-1][j], dp[i-1][j-v_i] + w_i)$

省メモリ
実装が楽

配列$dp[0..NV_{MAX}]$を用意
$For\ i\ =\ 1..NV_{MAX}$
　$dp[i] = \infty$
$dp[0]\ =\ 0$
$For\ i\ =\ 1..n$
　$For\ j\ =\ NV_{MAX}\ ..v_i$ (※逆順)
　　$dp[j]\ =\ \min(dp[j], dp[j-v_i] + w_i)$

考察＆解法　(C)荷物の重みが1000以下

- これが一番お馴染みかも知れない.
- $W_{MAX} = 1000$とおくと，重みの総和は高々 $N * W_{MAX}(= 20万)$ 以下.
- 重みに対して価値を最大化するDPができる.
- $dp[i][j]$:= i番目のアイテム$(i = 1..N)$まで使って，重みの和jを達成する価値の和の最大値

と定義すると，以下の漸化式を二重ループで計算すれば解ける

  - $dp[0][0] = 0, dp[0][1..NW_{MAX}] = -\infty$
  - $dp[i][j] = \max(dp[i-1][j], dp[i-1][j-w_i] + v_i$(但し$j - w_i \geqq 0$の時))

- 後は重みW以下で価値が最大のものを求めれば良い.
- さっきのスライドで説明した価値のDPとやり方はほぼ同じ

考察＆解法　(C)荷物の重みが1000以下

- これも重みの逆順に更新すれば,
  時間計算量は $O(N^2 W_{MAX})$　空間計算量は $O(NW_{MAX})$