ABC017 解説

解説スライド担当: 城下 慎也(@phidnight)

問題A – プロコン

問題概要

- 3 つの課題があり、それぞれ得点と何割できたかが与えられる。
- 合計得点を計算せよ。

- $10 \leqq$ 配点 $\leqq 990$
- $1 \leqq$ 割合 $\leqq 10$

解法

- 整数を読み込んで、課題ごとに (配点)*(割合)/10 を計算して合計得点を計算することでとくことができます。
- 読み込みには例えば C 言語だと入出力関数 (printf,scanfなど) を用います。
- 問題の制約上、入力も出力も整数であることが保証されています。

注意事項

- 「入力」と「出力」は整数値ですが、計算過程は整数値とは限りません。
- 例えば、(配点) * ((割合) / 10) のように除算を先に行ってしまうと実装次第では計算結果が変わってしまいます。
- 計算で浮動小数点型を用いて計算を行うこともできますが、その場合、最後の整数変換で誤差によって値がずれる可能性があることに注意してください(2.9999999... -> 2 にしてしまう)
- 小さな数 (0.1 とか) を足してから切り捨てると安全です。

問題B – choku 語

問題概要

- 文字列 S が choku 語(ch,o,k,u を組み合わせてできる文字列)であるか判定せよ。

- $1 \leqq |S| \leqq 50$

解法

- c,h,o,k,u 以外の文字が登場すれば、choku 語ではないことがわかります。
- 問題は ch が正しく登場しているかについてですが、これは、「c の直後に h があるか」と「h の直前に c があるか」をすべての c,h について判定することができます。
- 他にもいくつか判定する方法があります。

備考

- この問題は文字列の問題です。
- 1 文字ずつ読み込んだ場合、改行記号を文字列に含めてしまう場合があります。
- また、文字列長が与えられないので、終端符号か否かを判定する必要があります。

問題C – ハイスコア

問題概要

- 連続した整数を覆う区間がいくつか与えられます。
- 1 から M までのすべてが登場しないように区間の集合を選びます。
- 選んだ区間の値の合計を最大化してください。

- $1 \leqq N(\text{区間数}) \leqq 100{,}000$
- $1 \leqq M \leqq 100{,}000$

部分点解法 1 (30点)

- すべての区間の集合を考えます。
- 各数字について、その数字を覆う区間が集合にあるかどうかを判定します。
- どの区間にも含まれなかった数字があれば成立、そうでなければ不成立です。
- 全体の計算量は$O(N * M * 2^N)$ などになります。

部分点解法 2 (30+70点)

- すべての区間の組み合わせは指数オーダー存在するので、他の方針を考える必要があります。
- 現在選んでいる集合がどの数字を覆っているかを情報として保存する方針では、保存すべき状態数が2^M 個あり、これも大きいです。

部分点解法 2 (30+70点)

- 条件を満たす集合を考えると、条件を満たす集合では、「少なくとも 1 つの数字を覆っていない」という条件が成り立ちます。
- そこで、覆っていない数字を固定して考えてみます。
- 覆っていない数字を X としたとき、各区間について
  - 区間がX を覆っている場合、その区間は採用できない。
  - 区間がX を覆っていない場合、その区間は採用しても問題ない。
- 採用可能な区間すべてを採用しても X は覆われず、かつ採用自体には得点のデメリットはないので、すべてを採用するとそのX については最適解が求まります。

部分点解法 2 (30+70点)

- 先ほどの計算をすべての X について試すことで最適解を計算することができます。
- 計算量は、各 X (全部で M 通りある) について O(N) で判定できるので、全体で O(NM) となります。

満点解法 (30+70+1点)

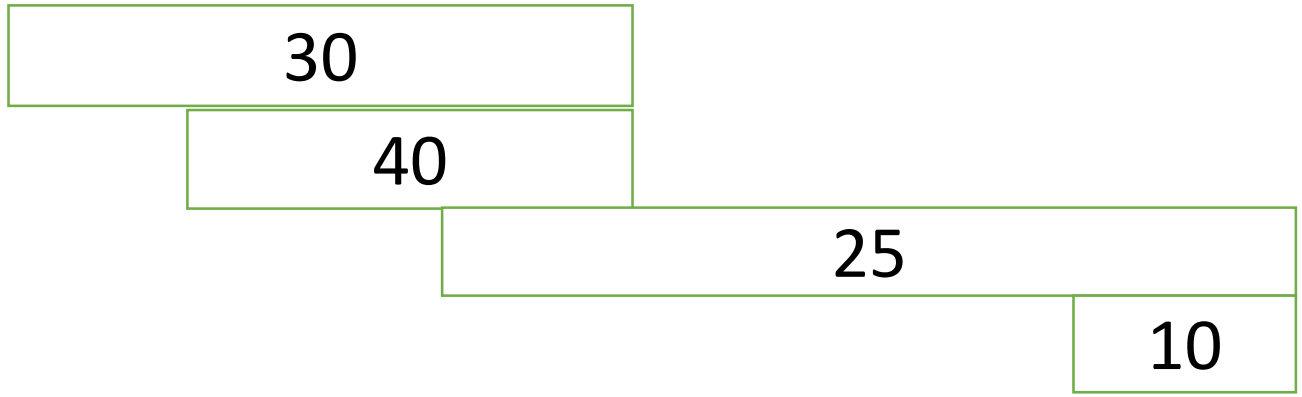
- 先の計算で、「覆っていない区間の合計得点を求める」のではなく、「全体の合計得点から覆っている区間の合計得点を引く」という方針で考えることにします。
- 覆っている区間の合計得点は、以下のことを実行できる配列,データ構造があれば実現できます。
  - ある連続した領域に同じ値を足す。
  - 特定の 1 要素の値を求める。

満点解法 (30+70+1点)

- そのようなデータ構造は、segment-tree を用いることで実現することができますが、今回は値の計算が最後にまとめて来ることから、競技プログラミング界隈では「いもす法」として知られている方法を用いることができます。
- 具体的には、配列 array の array[l] から array[r] までに v ずつ足したい場合には、1 つずつ足す代わりに array[l] だけに v を足し、array[r+1] に –v を足すようにします。
- 最後に array[i+1] = array[i] + array[i+1] を i = 1,2,…,M-1 について実行します。

満点解法 (30+70+1点)

- 例

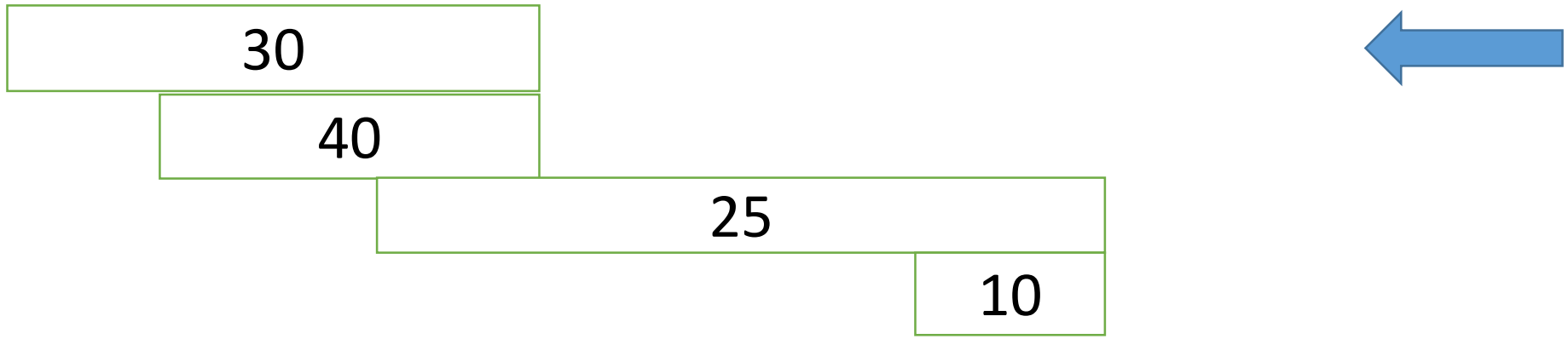


- Array

0	0	0	0	0	0	0	0

満点解法 (30+70+1点)

- 例

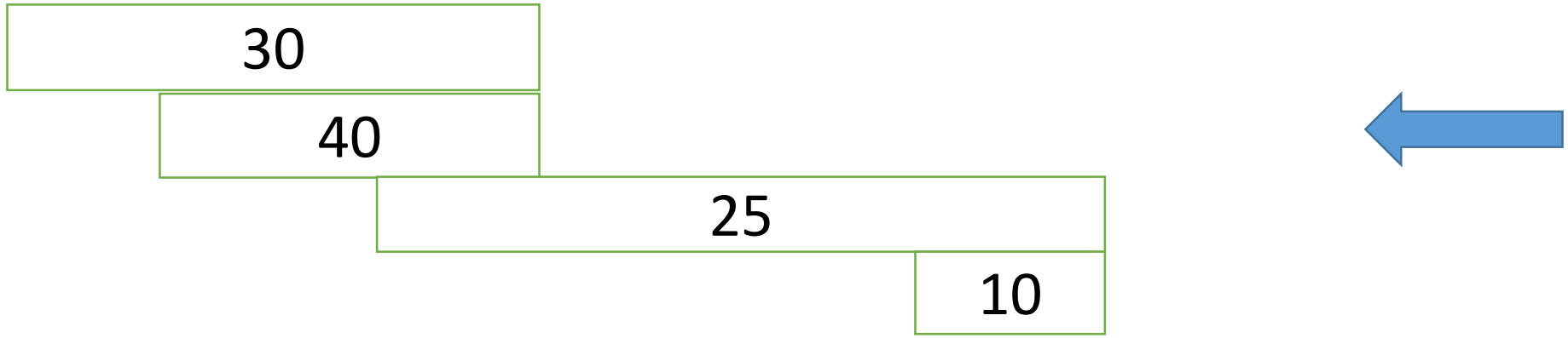


- Array

0	30	0	0	-30	0	0	0

満点解法 (30+70+1点)

- 例

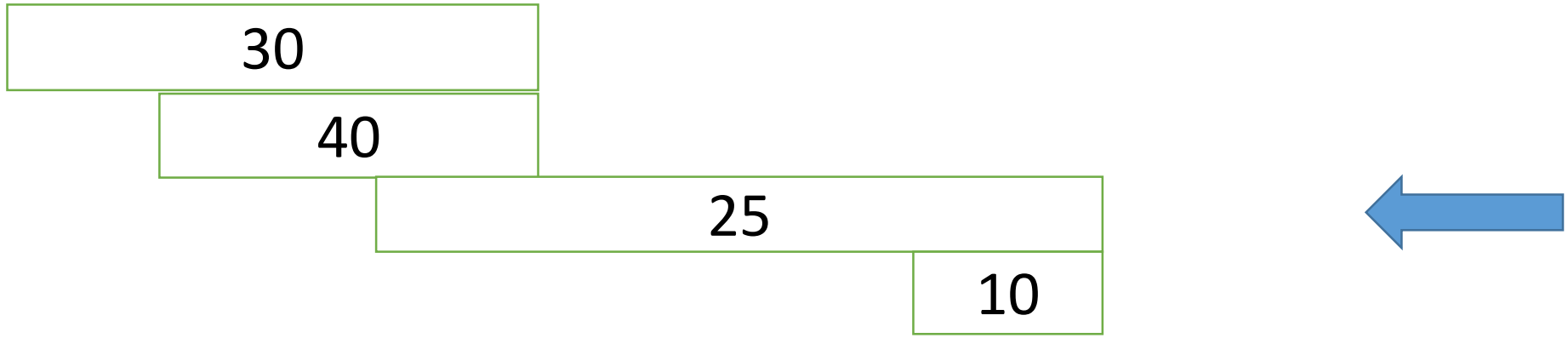


- Array

0	30	40	0	-70	0	0	0

満点解法 (30+70+1点)

- 例

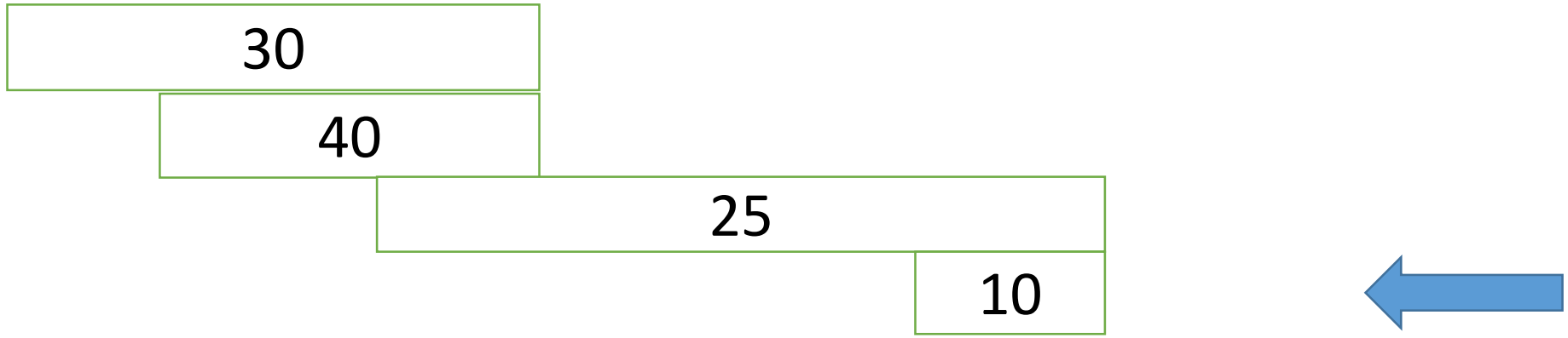


- Array

0	30	40	25	-70	0	0	-25

満点解法 (30+70+1点)

- 例

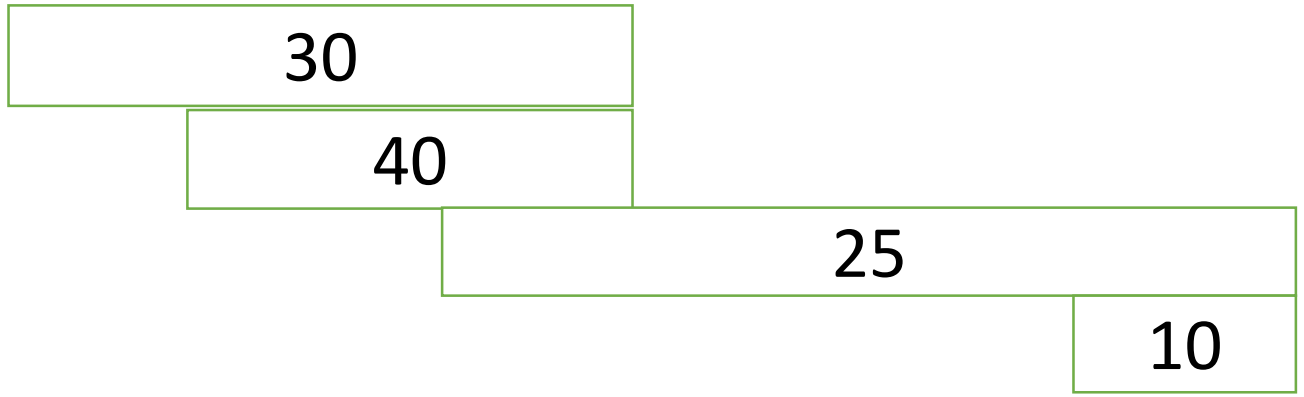


- Array

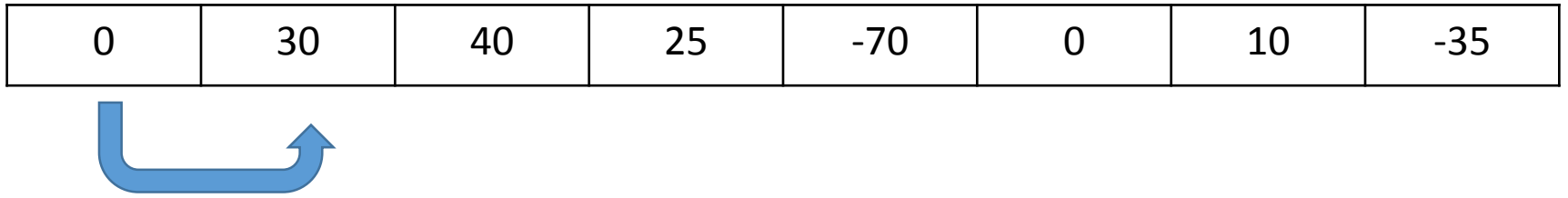
0	30	40	25	-70	0	10	-35

満点解法 (30+70+1点)

- 例

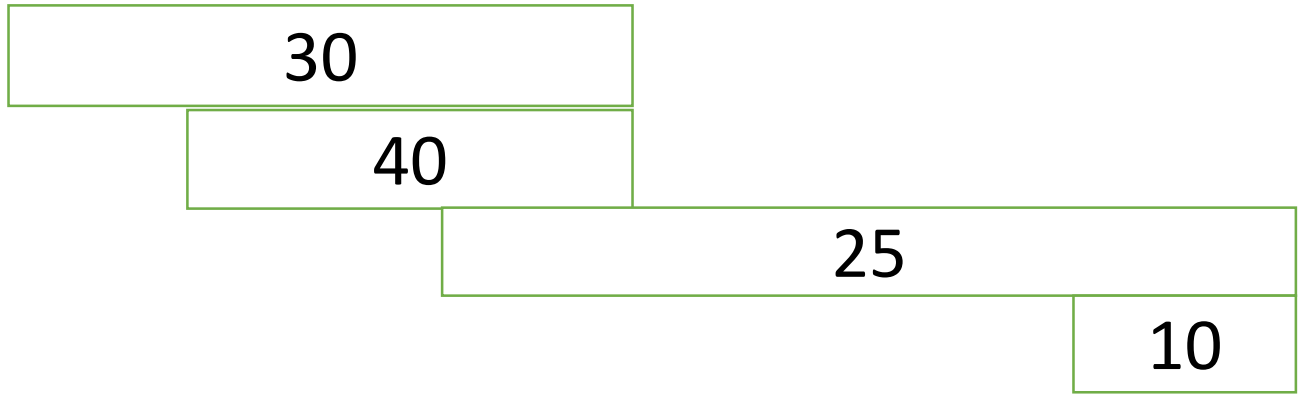


- Array

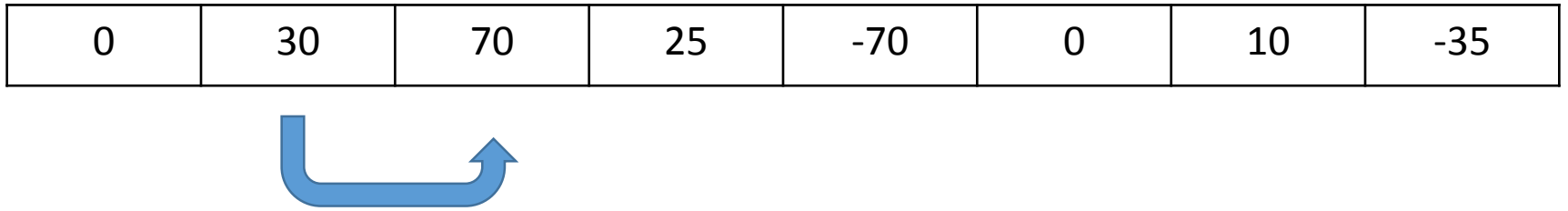


満点解法 (30+70+1点)

- 例

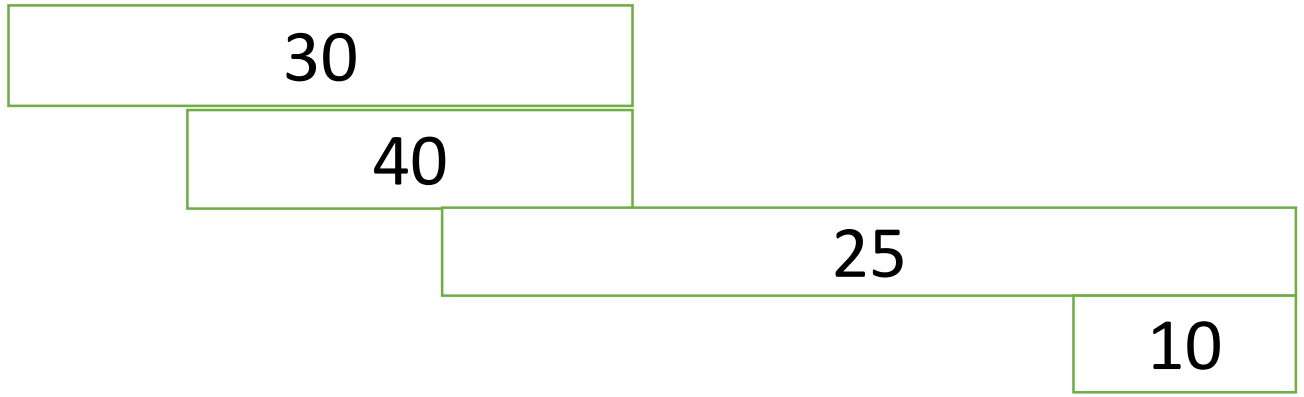


- Array

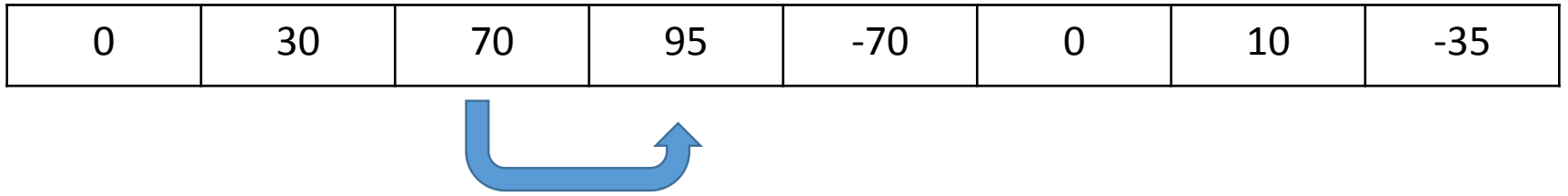


満点解法 (30+70+1点)

- 例

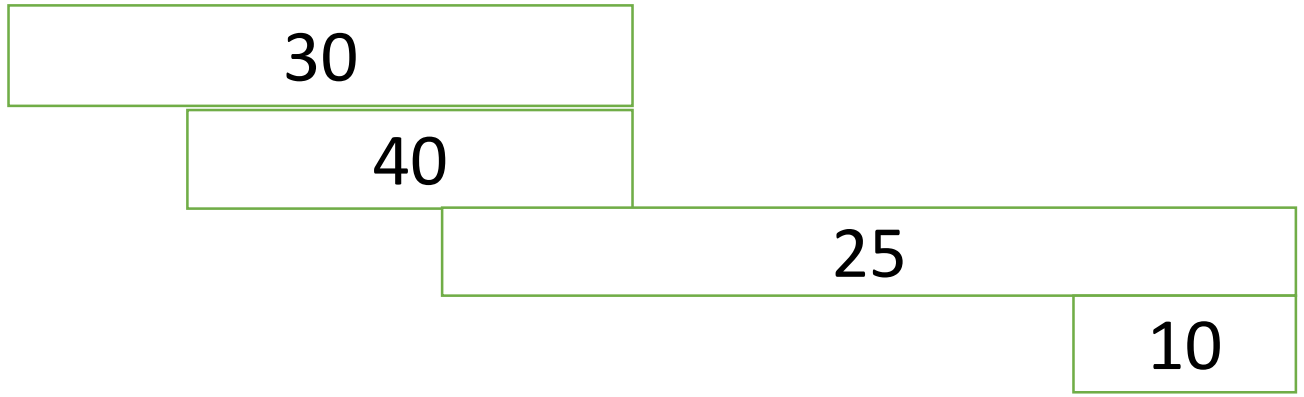


- Array

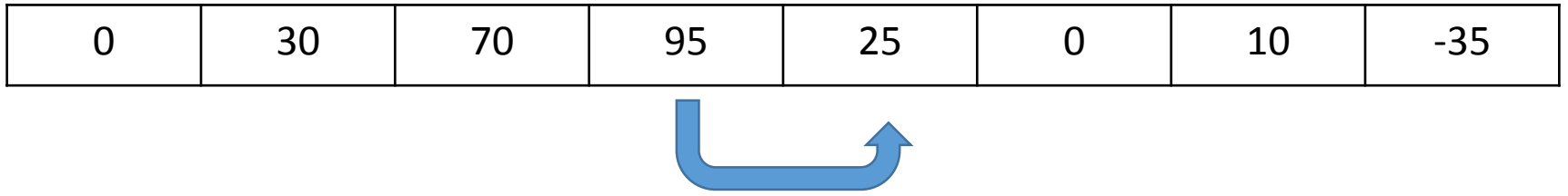


満点解法 (30+70+1点)

- 例

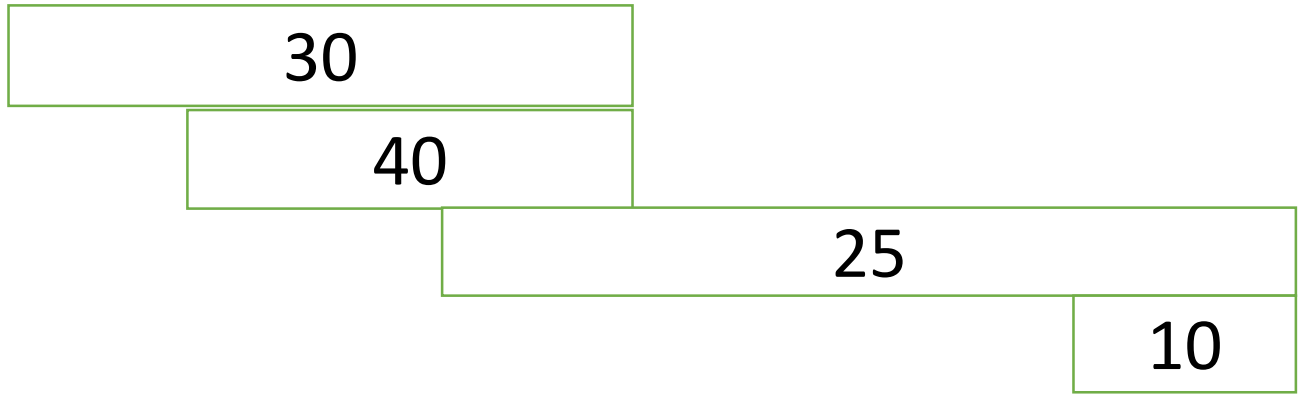


- Array

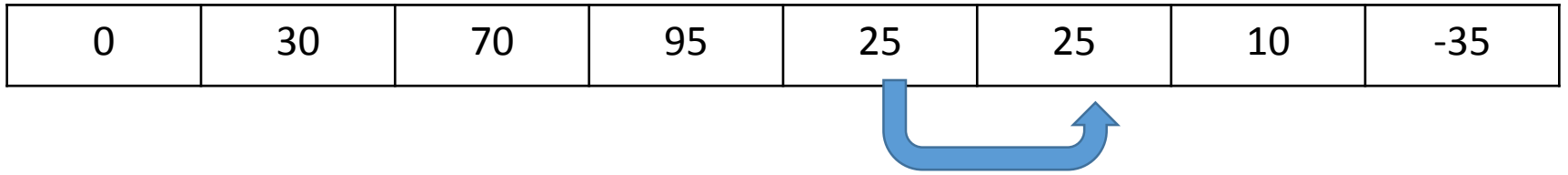


満点解法 (30+70+1点)

- 例

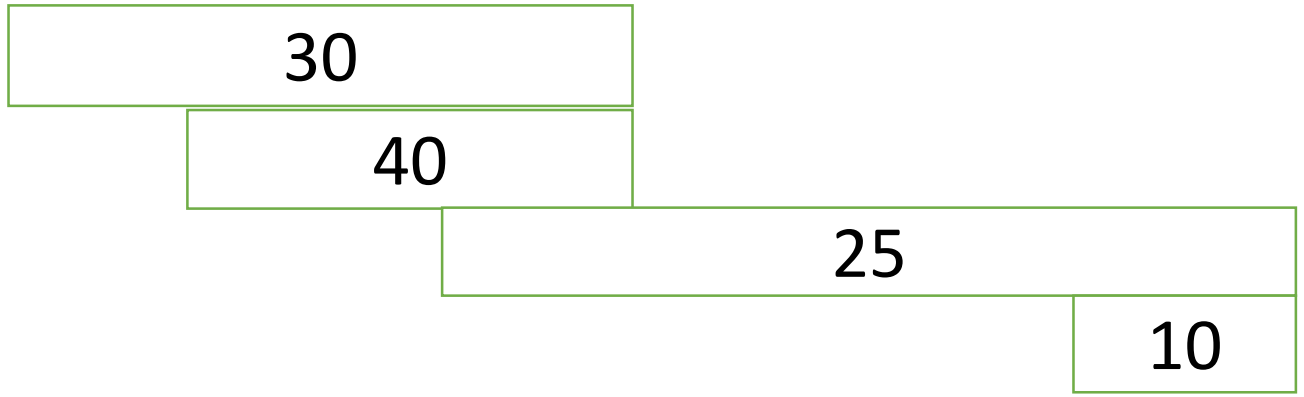


- Array

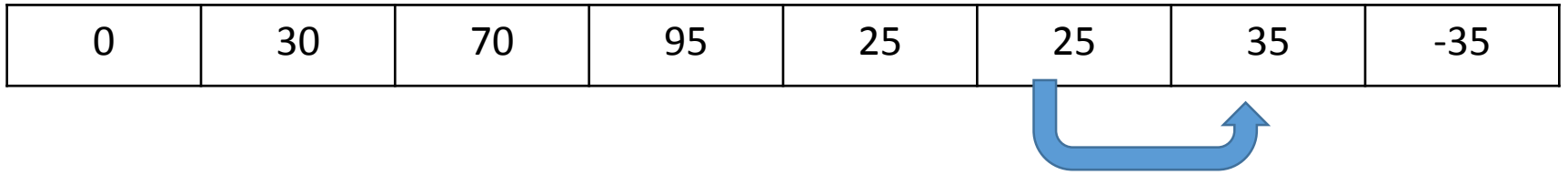


満点解法 (30+70+1点)

- 例

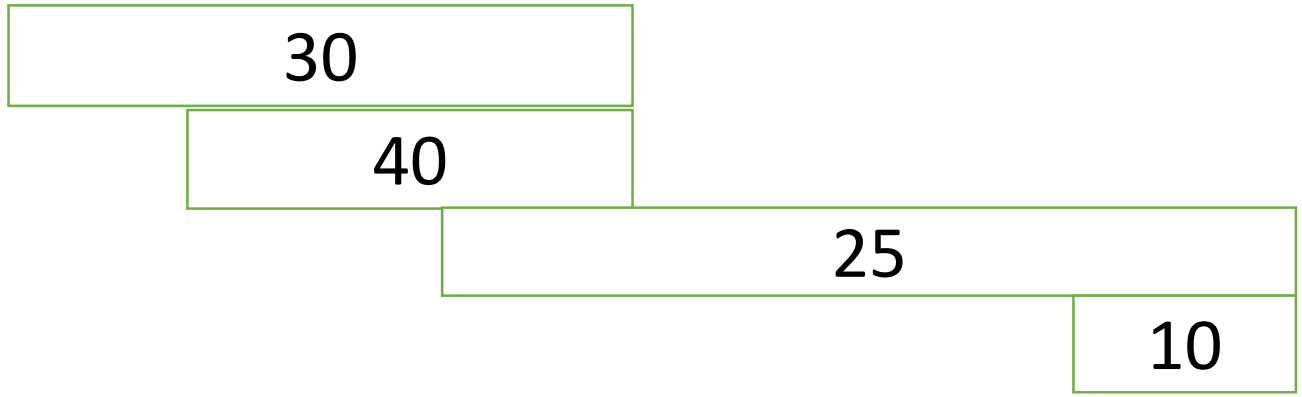


- Array

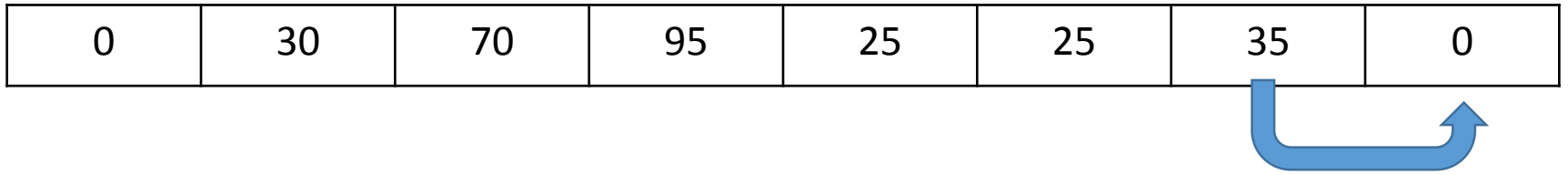


満点解法 (30+70+1点)

- 例



- Array

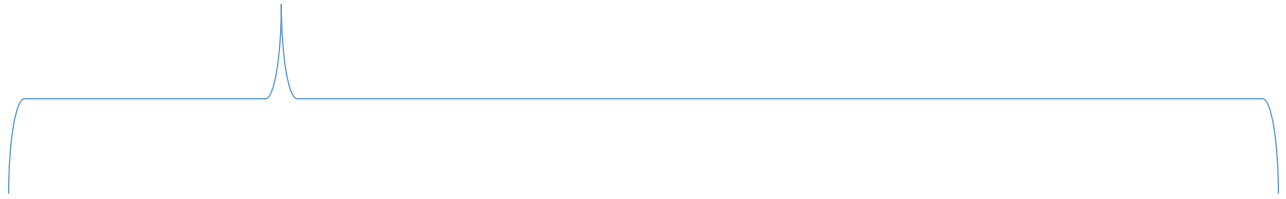


満点解法 (30+70+1点)

- 例

- この中の最小値(25) を全体 (105) から引いた 80 が答えとなります。

- Array



0	30	70	95	25	25	35	0

問題D – サプリメント

問題概要

- N個あるサプリメントを番号順に摂取する。
- 同じ味のサプリメントを同じ日に摂取することはできません。
- 考えられる組み合わせ数を計算してください。

- $1 \leqq N(\text{サプリメントの個数}) \leqq 100{,}000$
- $1 \leqq M(\text{色数}) \leqq 100{,}000$

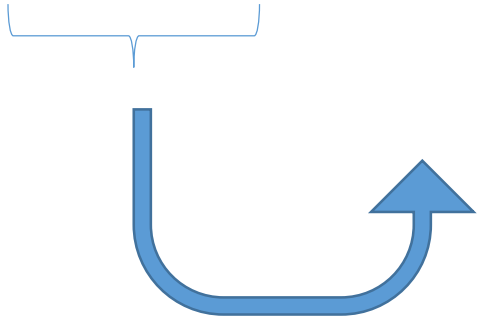
部分点解法 (30点)

- この問題は動的計画法を用いて解くことができます。
- dp[i] = (ある日の終了時点で i 番目のサプリメントまでを摂取する組み合わせ数) と置きます。
- このとき、各dp[i]については、i 以前の dp の値の組み合わせ「例で見るようにある連続したdpの値の合計」となります。
- これは、i まで食べる日の前日に j まで食べた場合に j 以前の食べ方がdp[j]通りあることから言えます。
- 例でもわかるように実は条件を満たすj はある連続した領域になります。

部分点解法 (30点)

- 例 (上はサプリメント番号と色,下は dp の値)

初日(何も食べてない)	サプリ1(色1)	サプリ2(色2)	サプリ3(色1)	サプリ4(色2)	サプリ5(色2)
1					

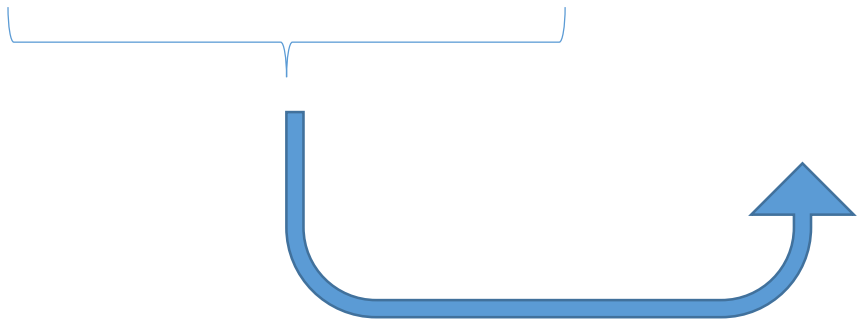


- この範囲ならば次のdp値に反映できる

部分点解法 (30点)

- 例 (上はサプリメント番号と色,下は dp の値)

初日(何も食べてない)	サプリ1(色1)	サプリ2(色2)	サプリ3(色1)	サプリ4(色2)	サプリ5(色2)
1	1				

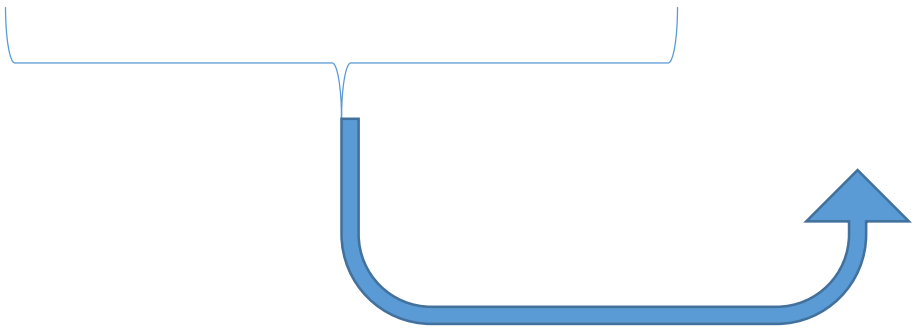


- この範囲ならば次のdp値に反映できる

部分点解法 (30点)

- 例 (上はサプリメント番号と色,下は dp の値)

初日(何も食べてない)	サプリ1(色1)	サプリ2(色2)	サプリ3(色1)	サプリ4(色2)	サプリ5(色2)
1	1	2			

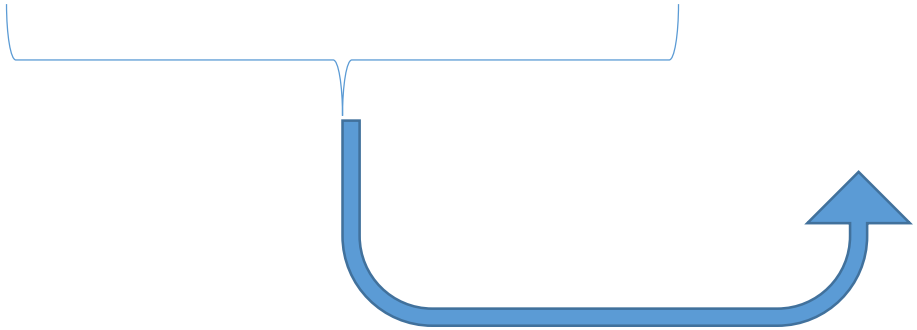


- この範囲ならば次のdp値に反映できる
  (dp[i] が「i 番目までを食べている」ことに注意)

部分点解法 (30点)

- 例 (上はサプリメント番号と色,下は dp の値)

初日(何も食べてない)	サプリ1(色1)	サプリ2(色2)	サプリ3(色1)	サプリ4(色2)	サプリ5(色2)
1	1	2	3		

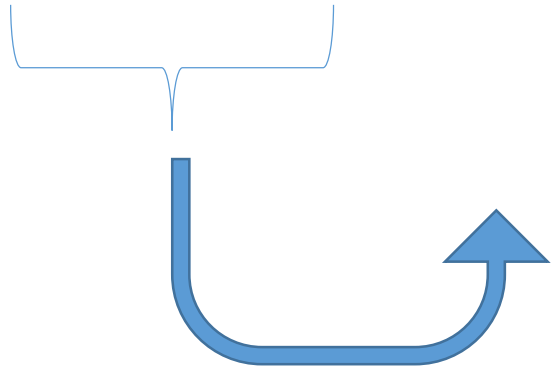


- この範囲ならば次のdp値に反映できる
  (dp[i] が「i 番目までを食べている」ことに注意)

部分点解法 (30点)

- 例 (上はサプリメント番号と色,下は dp の値)

初日(何も食べてない)	サプリ1(色1)	サプリ2(色2)	サプリ3(色1)	サプリ4(色2)	サプリ5(色2)
1	1	2	3	5	



- この範囲ならば次のdp値に反映できる
  (dp[i] が「i 番目までを食べている」ことに注意)

部分点解法 (30点)

- 例 (上はサプリメント番号と色, 下は dp の値)

初日(何も食べてない)	サプリ1(色1)	サプリ2(色2)	サプリ3(色1)	サプリ4(色2)	サプリ5(色2)
1	1	2	3	5	5

- このようにすれば、dp[N] が答えとなります。

部分点解法 (30点)

- 判定をする場合は、右端から配列に、その色が登場したという情報を順に格納していき 2 回登場したら終了するという方針で、各 i について O(N) で判定できるので、全体で O($N^2 + M$) となります。

満点解法

- 実は、先ほどのdpで足せる区間は、ある連続する区間であるということだけではなく、右端が右に行くにつれて左端が右にしか行かないということも成り立ちます。
- そこで、しゃくとり法を用いることができます。
- 各ステップについて、dp の区間を右に更新したら、同じ値が 2 回以上登場しなくなるまで左端を右に進めるようにします。
- dp の区間の合計も同様にしゃくとり法や累積和を用いて計算できます。
- O($N + M$) で解くことができます。