

## スーパーコン2012 予選・級認定課題

スーパーコン2012 審査委員会  
大阪大学・東京工業大学

### ナップサック問題とその変形

ナップサック問題というものがあります。これは、ある容量のナップサックと体積と価値の定められたいくつかの品物があるとき、容量を越えないように品物をナップサックに詰め込み、ナップサック内の品物の価値の和をどれだけ大きくできるかという問題です。今年の SuperCon の予選問題は、この「ナップサック問題」に文字列を取り入れた問題です。

以下、まず文字列についていくつかの定義を与えます。

- (1) 文字列  $A, B$  に対して  $A$  と  $B$  を連結した文字列を  $A + B$  で表すことにします。例えば、 $A = \text{"super"}, B = \text{"con"}$  に対して、

$$A + B = \text{"supercon"}, \quad B + A = \text{"consuper"}$$

となります。数値の演算記号  $+$  と異なり、ここでの文字列の連結記号  $+$  には順番があることに注意してください。

- (2) 文字列  $A$  に対して  $A$  の長さを  $|A|$  で表します。空文字列の長さは  $0$  です。例えば、 $A = \text{"supercon"}$  に対して、 $|A| = 8$  です。
- (3) 文字列  $A$  に対して  $A$  の  $i$  文字目を  $A(i)$  で表します。例えば、 $A = \text{"supercon"}$  に対して  $A(1) = s, A(2) = u, \dots, A(8) = n$  となります。
- (4) 文字列  $A, B$  に対して、辞書式順序で  $A$  が  $B$  より大きくないということを  $A \leq B$  と表記します。つまり C 言語の関数 `strcmp` での大小比較と同様です。厳密には、 $A \leq B$  であることの条件は、

- (a)  $A(i) \neq B(i)$  であるような最小の  $i (1 \leq i \leq \min(|A|, |B|))$ 、 $\min(X, Y)$  は  $X, Y$  のうち大きくない方が存在して、ASCII コードの比較 (アルファベット順) で  $A(i) < B(i)$  であるか、
- (b)  $A(i) \neq B(i)$  であるような  $i$  が存在せず  $|A| \leq |B|$  であるとしています。

例えば、(a) の場合の例は

$$\text{"con"} \leq \text{"super"}, \text{"superaon"} \leq \text{"superconconcon"}, \text{"superabcdefg"} \leq \text{"superb"}$$

などです。上の 2 つ目の例  $A = \text{"superaon"}$  と  $B = \text{"superconconcon"}$  の場合、最初から 5 文字までは同じで、6 文字目で初めて違いがあるので、違いの出る最小の  $i$  は 6 となります

( $A(6) \neq B(6)$ )。そして、6文字目のASCIIコードを比較すると、 $A(6)=a$ は $B(6)=c$ より小さいので、“superaon” $\leq$ “superconconcon”となります。この場合は7文字目以降がどんな文字列でも、またどれだけ長くても比較には影響しないことに注意してください。また、(b)の場合の例は

$$\text{“super”} \leq \text{“supercon”}, \text{“super”} \leq \text{“super”}, (\text{空文字列}) \leq \text{“supercon”}$$

などです。

### SuperCon2012 予選・認定問題

今回の予選・認定問題では、それぞれ文字列  $W$  および  $N$  個の文字列  $w_0, w_1, \dots, w_{N-1}$  と整数  $v_0, v_1, \dots, v_{N-1}$  が与えられます。 $W$  はナップサックの容量に対応し、 $v_i$  は  $i$  番目の文字列  $w_i$  の“価値”です。問題は、 $N$  個の文字列から何個かを選んで、以下の条件を満たすように（辞書式順序で  $W$  より大きくならないように）結合し、なるべく価値の合計が高い文字列を作るといいます。つまり、これらの入力に対して、

$$w_{S_0} + w_{S_1} + \dots + w_{S_{k-1}} \leq W$$

( $k \leq N$ ) および

$$0 \leq S_0 < S_1 < \dots < S_{k-1} < N$$

を満たすような整数列  $S = (S_0, S_1, \dots, S_{k-1})$  を選んで、対応する整数  $v_i$  の和

$$V = v_{S_0} + v_{S_1} + \dots + v_{S_{k-1}}$$

のとりうる最大値を求めてください。なお、整数列  $S$  は長さ 0 でも構いません。(空文字列)  $\leq W$  なのでそのような選び方はいつでも制約を満たし、そのときには  $V = 0$  ということになります。

各級の認定と予選の問題については以下の制約に従うものとします。

#### 問 A (スーパーコン 3 級認定問題 2012 年度版)

$N = 10$
$1 \leq  w_i  \leq 10$
$1 \leq  W  \leq 100$
$-1000000 \leq v_i \leq 1000000$

ただし、 $w_i, W$  は小文字のアルファベット (a, b, ..., z) からなる文字列。

ヒント：この問題は可能な文字列の結合全ての総当りが可能です。上記の条件を満たす問題例 10 題に対して全て、 $V$  の最大値を正しく求めた場合にスーパーコン 3 級を認定します。

#### 問 B (スーパーコン 2 級認定問題 2012 年度版)

$$\begin{aligned} 1 &\leq N \leq 1000 \\ 1 &\leq |w_i| \leq 1000 \\ 1 &\leq |W| \leq 10000 \\ -1000000 &\leq v_i \leq 1000000 \end{aligned}$$

ただし、 $w_i, W$  は  $a$  のみからなる文字列。

ヒント：この問題は可能な文字列の結合全ての総当りはできません。上記の条件を満たす問題例 10 題に対して全て、 $V$  の最大値を正しく求めた場合にスーパーコン 2 級を認定します。

### 問 C (スーパーコン予選問題 兼 1 級認定問題 2012 年度版)

$$\begin{aligned} 1 &\leq N \leq 1000 \\ 1 &\leq |w_i| \leq 10 \\ 1 &\leq |W| \leq 10000 \\ -1000000 &\leq v_i \leq 1000000 \end{aligned}$$

ただし、 $w_i, W$  は小文字のアルファベット  $(a, b, \dots, z)$  からなる文字列。

上記の条件を満たす問題例 10 題に対して全て、 $V$  の最大値を正しく求めた場合にスーパーコン 1 級を認定します。

### 注意

#### • 作成するプログラム

- プログラムは入出力の部分を規定した雛形プログラムをもとに、その一部を修正する形で作って下さい。作成の際には「変更可能」とコメントされている範囲以外は変更・削除しないようにして下さい。したがって、ここで指定された入力、出力以外のものを期待したプログラムは審査対象から除外されます。その他は適宜変更や追加して構いません。また、標準ライブラリ関数等を宣言して使用しても構いません。
- 提出するプログラムはチーム名をファイル名とする単一ファイルとして下さい。
- プログラムは C 言語で記述して下さい。詳細は以下の通りです。
  - \* プログラムは C 言語規格 (ANSI C や C99 など) に準拠する C 言語で記述して下さい。(インラインアセンブラの使用は禁止します)
  - \* int は 32 ビット、long long は 64 ビットを仮定します。long のビット幅は環境により異なるので、64 ビットデータを扱いたい場合は long ではなく long long を使って下さい。(stdint.h をインクルードして使える int64\_t などを使うとさらに確実です)
  - \* Linux 上の gcc ver 3.3.3 を使用してコンパイルし、Linux 上で実行します。この Linux は 64 ビット x86\_64 版です。

- \* 移植性の問題 (例えばエンディアンの違い) によるトラブルには対処しません。
- \* 文字列の比較に対しては、string.h をインクルードして strcmp や strncmp などを使って構いません。

- 審査方法 (スーパーコン認定に関して)

- 応募プログラムをコンパイルし、10 題の問題例に対して各々 1 分間の制限時間で実行し、すべての問題例で制限時間内に正確な答を出している場合に合格とします。
- 審査環境におけるメモリはおよそ 4 ギガバイトです。

- 審査方法 (予選選抜に関して)

- 応募プログラムをコンパイルし、10 題程度の問題例に対して各々 1 分間の制限時間で実行します。
- 審査環境におけるメモリはおよそ 4 ギガバイトです。
- 制限時間内に正確な答を出している問題例の個数順をもとに順位を決めます。
- 同順位のチームに対しては、正解を出した計算の計算時間の合計を用いて順位を決めます。(万が一、合計計算時間に優位差の無い場合には、合わせて提出されるプログラムの説明をもとに工夫度を評価して順位を決めます)
- 以上の順位付けのもとで、東西上位 10 チームずつを本選出場候補チームとして選びます。ただし、1 校からは最大 2 チームしか本戦出場できません。本選出場候補チームには、同時にスーパーコン 1 級も授与します。

## 詳細説明

- 問題例の与え方

問題例は以下のように与えられます。

$N$	
$W$	
$v_0$	$w_0$
$v_1$	$w_1$
...	...
$v_{N-1}$	$w_{N-1}$

例えば、 $N = 3$ ,  $W = \text{jihgfedcb}$ ,  $(w_0, w_1, w_2) = (\text{jih}, \text{adgfi}, \text{ghe})$ ,  $(v_0, v_1, v_2) = (10, -3, 5)$

に対する入力は次のようになります。

3	
jihgfedcb	
10	jih
-3	adgfi
5	ghe

この場合、 $v_1 = -3$  (負) なので、一見  $w_1$  は選ばない方がよいように思えますが、 $v_2 = 5$  なので  $w_2$  も選ぶことにより  $V$  を  $w_0$  だけを選ぶときの値  $V = v_0 = 10$  よりも大きな値  $V = 10 - 3 + 5 = 12$  にすることができ、 $V = 12$  が正解となります。 $w_1$  を選ばずに  $w_0$  と  $w_2$  だけを選んで  $V = v_0 + v_2 = 10 + 5$  としたいところですが、 $w_0 + w_2 = jihghe$  となって  $w_0 + w_2 \leq W$  の条件を満たさなくなってしまうので、そのような選び方はできないことにも注意してください。また、 $N = 3$ ,  $W = jihgfedcb$ ,  $(w_0, w_1, w_2) = (jih, adgfi, ghe)$ ,  $(v_0, v_1, v_2) = (10, -3, 2)$  の場合には、 $w_0$  だけを選ぶのが  $V$  を最大にするので、 $V = 10$  が正解です。

- 雛形プログラムについて

雛形プログラムでは、上記の入力問題例のデータをそれぞれ以下のように `int` 型の  $N$ , `char` 型の配列  $W$ 、`char` 型の 2 次元配列  $w$ , `int` 型の配列  $v$  に格納します。それらを用いて「変更可能」とコメントされている範囲にプログラムを書いてください。

```

N = 3;
W = "jihgfedcb";

w[0] = "jih";
w[1] = "adgfi";
w[2] = "ghe";

v[0] = 10;
v[1] = -3;
v[2] = 5;

```