

8. 並び替え

8.1 単純交換法

隣り合う要素を比較し、必要であれば交換する作業を繰り返して並び替えます。この方法では、値の小さいデータが交換のたびに上に動いてくるようにみえます。ですから、泡が液体中を上にあがってくる様子にたとえて、**バブルソート(bubble sort)**、**泡立ちソート**等と呼ばれます。単純交換法では、同じ値であれば順序の逆転がない安定なソートになります。

```
void bubbleSort(char *table[], int N) {
    char *T;
    for(int i=0; i<N-1; i++) {
        for(int j=N-1; j>i; j--)
            if(strcmp(table[j-1], table[j])>0) {
                T=table[j-1];table[j-1]=table[j];table[j]=T;
            }
    }
}
```

8.2 単純選択法

着目する位置と、それより後方の値を比較し、最小値の位置のデータと交換します。降順にするには最大値を求めます。単純選択法は、単純交換法に比べ、データを交換するコストが非常に少なくなります。離れた位置を交換するため同一の値で順序の逆転を許す不安定なソートになります。

```
void selectionSort(char *table[], int N) {
    char *min, *T;
    for(int i=0; i<N-1; i++) {
        min = table[i];int id=i;
        for(int j=i+1; j<N; j++)
            if(strcmp(table[j], min)<0) {
                id=j; min=table[j];
            }
        if(id!=i) {
            T=table[i];table[i]=table[id];table[id]=T;
        }
    }
}
```

8.3 単純挿入法

着目する位置と、それより後方の値を比較し、最小値の位置のデータと交換します。**単純挿入法(straight insertion sort)**とは、着目した要素を適当な位置に挿入する作業を繰り返してソートします。トランプ等を昇順に並べる方法に似ており、**シャトルソート(shuttle sort)**とも呼ばれます。

```
void insertSort(char *table[], int N) {
    for(int i = 1; i < N; i++) {
        char *Tmp = table[i];int j;
        for(j = i; j > 0 && strcmp(table[j-1], Tmp) >0; j--) table[j] = table[j-1];
        table[j]=Tmp;
    }
}
```