

Need help for a code in "Numerical Recipes in C++"

Source: <http://sci.tech-archive.net/Archive/sci.math.num-analysis/2005-01/0009.html>

From: Mok-Kong Shen (mok-kong.shen_at_t-online.de)

Date: 12/31/04

Date: Fri, 31 Dec 2004 15:49:40 +0100

The code of the heapsort algorithm as given in books by e.g. Sedgewick deals with arrays with indices in [1..n]. For programming in languages like C, one would like to sort arrays with indices in [0..n-1]. I have found sofar only one book that gives code for the latter case, namely "Numerical Recipes in C++" by W. H. Press et al., which itself refers to Sedgewick. (Note that a companion book by the same authors for C has code for indices in [1..n], which is a bit odd in that context.)

I adapted the code in the C++ book to C for sorting arrays of 'int' with indices in [0..n-1]. However, the resulting code (see attachment below) does not always function correctly. As concrete examples, of 8 randomly chosen cases tested for an 'int' array of size 5 consisting of [0..4] the following input

```
2 1 3 0 4
```

was wrongly sorted to

```
0 1 2 4 3
```

while the following inputs

```
3 0 4 2 1
```

```
4 1 0 2 3
```

```
4 3 0 1 2
```

```
3 0 1 4 2
```

```
2 3 1 4 0
```

```
2 3 4 0 1
```

```
1 0 4 3 2
```

were all processed correctly.

Since I have only the book but not the corresponding original

software, I couldn't be sure whether this is due to my coding error in adapting the code from the text of the book or whether this is due to an error of the authors of the book (though I have carefully checked my coding several times to be quite sure that it corresponds in essence to the code given in the book). If there is some person who happens to possess the original software of the C++ book, I should appreciate very much to know a confirmation/refutation of my computing result above.

M. K. Shen

```
void siftdown(int a[], int l, int r)
{ int j,jold,v;
  v=a[l];
  jold=l;
  j=l+1;
  while (j<=r)
  { if (j<r && a[j]<a[j+1]) j++;
    if (v>=a[j]) break;
    a[jold]=a[j];
    jold=j;
    j=2*j+1;
  }
  a[jold]=v;
}

void hpsort(int a[], int n)
{ int i,t;
  for (i=n/2-1; i>=0; i--) siftdown(a,i,n-1);
  for (i=n-1; i>0; i--)
  { t=a[0]; a[0]=a[i]; a[i]=t;
    siftdown(a,0,i-1);
  }
}
```