SANGFOR 深信服科技 深信服校园招聘c/c 软件开发B卷 企业提供原题 00:00:45

2/3 [编程题]堆排序

时间限制：C/C++ 1秒，其他语言2秒
空间限制：C/C++ 32M，其他语言64M

函数heap_sort使用堆排序方法对数组arr进行排序，排序后数据为降序。相关代码如下，请补充缺失部分。

```
void heap_arrange(int arr[], int cur, int cnt)  //调整为小顶堆
{
    int heaptop_val = arr[cur]; //堆顶的值
    while (cur < cnt) {
        int left = 2 * cur + 1;
        int right = 2 * cur + 2;
        int min = -1;
        int min_val = _______;
        if (left < cnt && arr[left] < min_val) { //检查是否比左节点大
            min = left;
            min_val = arr[left];
        }
        if (right < cnt && arr[right] < min_val) {//检查是否比右节点大
            min = right;
        }
        if (min == _______)
            break;
        arr[cur] = _______;
        cur = _______;
    }
    arr[cur] = _______;
}
void heap_sort(int arr[], int cnt)
{
    int i;
    for (i = cnt / 2 - 1; i >= 0; --i) {
        heap_arrange(arr, i, cnt);
    }
    for (i = cnt - 1; i > 0; --i) {
        int tmp;
        tmp = arr[0];
        arr[0] = arr[i];
        arr[i] = tmp;
        heap_arrange(arr, 0, i);
    }
}
```

输入描述:

> 第一行为数据个数 第二行为输入数据

输出描述:

> 排序过程的中间数据，及已经排好序的数据

输入例子1:

输出例子1:

```
origin:
100
 32   3
  6  24  86  23
 90  78   3

make heap:
  3
  6   3
 78  24  86  23
 90 100  32

sort i=9
 32
  6   3
 78  24  86  23
 90 100   3

  3
  6  23
 78  24  86  32
 90 100   3

sort i=8
100
  6  23
 78  24  86  32
 90   3   3

  6
 24  23
 78 100  86  32
 90   3   3

sort i=7
 90
 24  23
 78 100  86  32
  6   3   3

 23
 24  32
 78 100  86  90
  6   3   3

sort i=6
 90
 24  32
 78 100  86  23
  6   3   3

 24
```

```
  6   3   3

sort i=5
 86
 78  32
 90 100  24  23
  6   3   3

 32
 78  86
 90 100  24  23
  6   3   3

sort i=4
100
 78  86
 90  32  24  23
  6   3   3

 78
 90  86
100  32  24  23
  6   3   3

sort i=3
100
 90  86
 78  32  24  23
  6   3   3

 86
 90 100
 78  32  24  23
  6   3   3

sort i=2
100
 90  86
 78  32  24  23
  6   3   3

 90
100  86
 78  32  24  23
  6   3   3

sort i=1
100
 90  86
 78  32  24  23
  6   3   3

100
 90  86
 78  32  24  23
  6   3   3
```

```
100
 90  86
 78  32  24  23
  6   3   3
```

C++11(clang++ 3.9) 重置 自测

```c
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
static void heap_arrange(int arr[], int cur, int cnt);

static int heap_verify(int arr[], int cnt)
{
    int i;
    for (i = 0; i < cnt / 2; ++i) {
        int lhs = 2 * i + 1;
        int rhs = 2 * i + 2;

        if (lhs < cnt && arr[i] > arr[lhs]) {
            fprintf(stderr, "arr[%d]:%d > arr[%d]:%d\n", i, arr[i], lhs, arr[lhs]);
            return -1;
        }
        if (rhs < cnt && arr[i] > arr[rhs]) {
            fprintf(stderr, "arr[%d]:%d > arr[%d]:%d\n", i, arr[i], rhs, arr[rhs]);
            return -1;
        }
    }
    return 0;
}

static void heap_print(int arr[], int cnt)
{
    int layer = 0, num = 0;
    for (layer = 0; num < cnt; ++layer) {
        int i = 0;
        for (i = 0 ; i < (1 << layer) && num < cnt ; ++i)
            printf("%3d ", arr[num++]);
        printf("\n");
    }
}
static void heap_sort(int arr[], int cnt)
{
    int i;

    printf("origin:\n");
    heap_print(arr, cnt);
    // 建堆
    for (i = cnt / 2 - 1; i >= 0; --i) {
        heap_arrange(arr, i, cnt);
    }
    printf("make heap:\n", i);
    heap_print(arr, cnt);
    assert(heap_verify(arr, cnt) == 0);
```

```
50 ····tmp = arr[0];
51 ····arr[0] = arr[i];
52 ····arr[i] = tmp;
53 ····printf("sort i=%d\n", i);
54 ····heap_print(arr, cnt);
55 ····heap_arrange(arr, 0, i);
56 ····heap_print(arr, cnt);
57 ····assert(heap_verify(arr, i) == 0);
58 ····}
59 ····printf("sorted:\n");
60 ····heap_print(arr, cnt);
61 }
62 static int input(int **arr, int *size)
63 {
64 ····int i;
65 ····int ret;
66 
67 ····ret = fscanf(stdin, "%d\n", size);
68 ····if (ret != 1)
69 ····return -1;
70 ····*arr = (int *)malloc(sizeof(int) * (*size));
71 ····for (i = 0; i < *size; ++i) {
72 ····fscanf(stdin, "%d ", &(*arr)[i]);
73 ····}
74 ····return 0;
75 }
76 
77 int main(int argc, char *argv[])
78 {
79 ····int *arr = NULL;
80 ····int cnt = 0;
81 ····int i;
82 
83 ····if (input(&arr, &cnt) < 0) {
84 ····fprintf(stderr, "input error\n");
85 ····return 0;
86 ····}
87 ····heap_sort(arr, cnt);
88 ····return 0;
89 }
90 //··调整为小顶堆
91 static void heap_arrange(int arr[], int cur, int cnt) ··//调整为小顶堆
92 {
93 ····int heaptop_val = arr[cur]; ·//堆顶的值
94 ····while (cur < cnt) {
95 ····int left = 2 * cur + 1;
96 ····int right = 2 * cur + 2;
97 ····int min = -1;
98 ····int min_val = ______;
99 ····if (left < cnt && arr[left] < min_val) { ·//检查是否比左节点大
100 ····min = left;
101 ····min_val = arr[left];
102 ····}
103 ····if (right < cnt && arr[right] < min_val) {//检查是否比右节点大
104 ····min = right;
105 ····}
```

```
108 ·····arr[cur]·=·_____;
109 ·····cur·=·_____;
110 ····}
111 ····arr[cur]·=·_____;
```

提交运行 提前交卷 下一题

* 交卷即可查看全部答案和解析，完成所有题目有机会获得企业内推

收藏本题 标记一下 场外求助 提交结果有问题?

收起答题卡 编程题3道

1 2 3