

哈尔滨工业大学

2010 年春季学期数据结构与算法 试卷 A 参考答案

一、填空题（每空 1 分，共 15 分）

1. $\log i = \log j$ 2. CDBGFEA 3. $n-1$ 4. $O(n^2)$ 5. $abc+*d-$ 6. 5 18 7. $i(i+1)/2+j-1$
8. 16 18 19 20 30 22 9. 多路归并、I/O 并行处理 初始归并段产生 10. 1,4,2,3
11. 3 12. 非负

二、选择题（每小题 1 分，共 15 分）

1. A 2. C 3. A 4. A 5. A
6. C 7. A 8. C 9. B 10. A
11. D 12. C 13. C 14. D 15. B

三、简答题（每题 10 分，共 20 分）

1. 按数组元素有序的一维数组一定是堆。（4 分）

非升序数组一定是最小堆为例说明如下：假设非升序数组为 K_1, K_2, \dots, K_n ，则满足 $K_1 \leq K_2, \leq \dots \leq K_n$ ，则一定满足： $K_i \leq K_{2i}$ 且 $K_i \leq K_{2i+1}$ ，即满足最小堆的定义。同理可知，非降序数组一定是最大堆。因此，按数组元素有序的一维数组一定是堆。（6 分）

2. 二叉平衡树为 48, 40, 80, 22, 45, 78（3 分）

前序：48, 40, 22, 45, 80, 78（3 分）

右旋转函数：

```
void R_Rotate(BSTree &p)
{
    lc = p->lchild;
    p->lchild = lc->rchild;
    lc->rchild = p;
    p = lc;
}
```

四、算法设计（每题 10 分，共 20 分）

1. 本算法不要求将整个记录进行排序，而只进行查找第 j 个记录。

基本思想：改进划分算法，是一次划分将基准元素定位于 k ，如果 $k=j$ ，则找到第 j 小的元素；否则，递归地在 k 的左边或右边进行划分，直到 $k=j$ 为止。

算法如下：

```
int Search(int A[], int n, int j)
{
    s = 1; t = n;
    k = Partition(A, s, t);
    while (k != j)
        if (k < j) k = Partition(A, k + 1, t);
        else k = Partition(A, s, k - 1);
}
```



```
    return A[j];
}
int Partition(int A[], int low, int high)
{
    i = low; j = high; pivot = A[low];
    while (i < j)
    {
        while (A[j] >= pivot && i < j) j--;
        if (i < j)
            A[i++] = A[j];
        while (A[j] < pivot && i < j) i++;
        if (i < j)
            A[j--] = A[i];
    }
    A[i] = pivot;
    return i;
}
2. int visited[MAXSIZE]; //指示顶点是否在当前路径上
int exist_path_DFS(ALGraph G, int i, int j)
{
    if (i == j) return 1; //i就是j
    else
    {
        visited[i] = 1;
        for (p = G.vertices[i].firstarc; p; p = p->nextarc)
        {
            k = p->adjvex;
            if (!visited[k])
                return exist_path(k, j); //i下游的顶点到j有路径
        }
        return 0;
    }
}
//exist_path_DFS
```