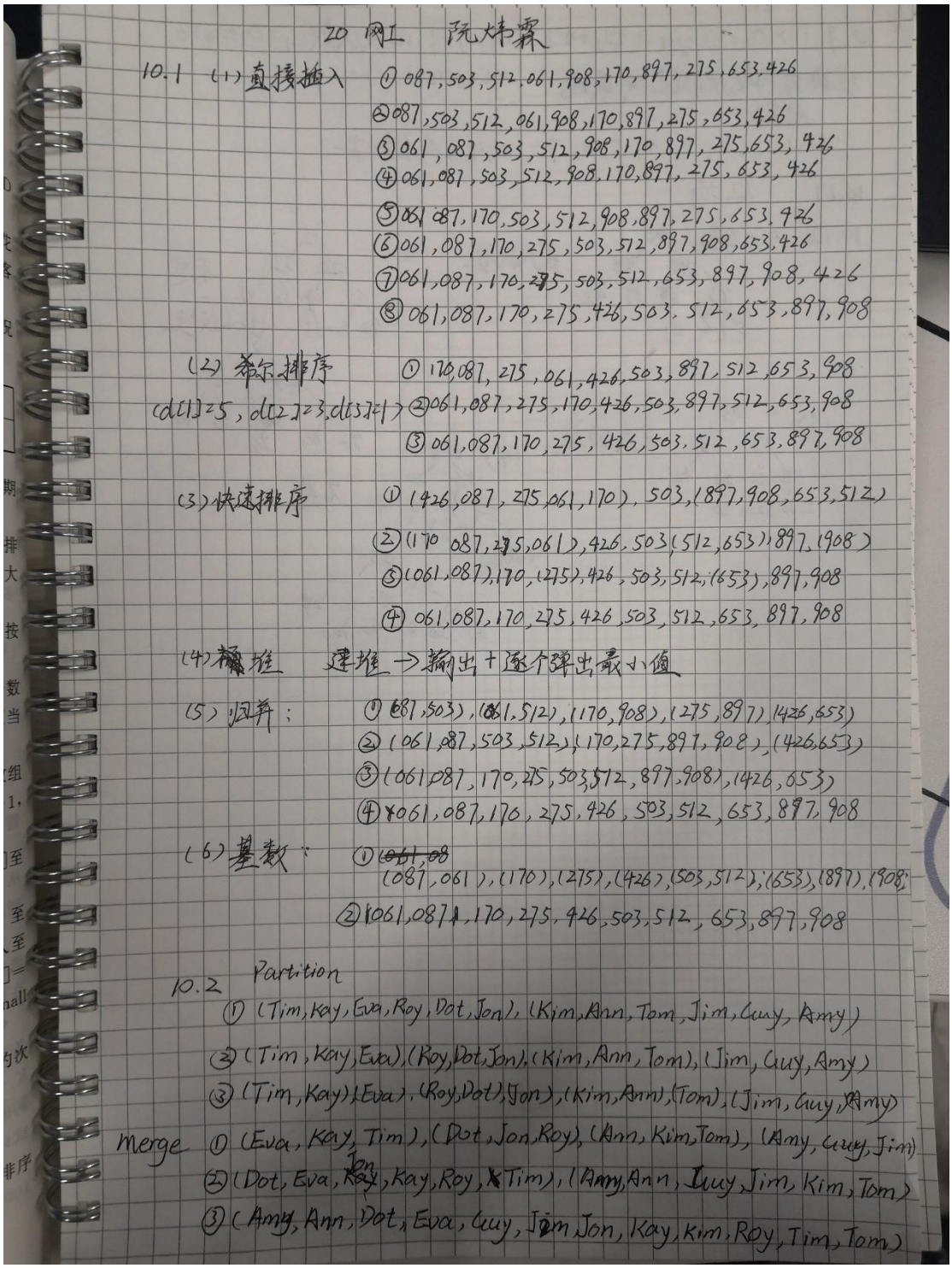
数据结构 第十章 作业　阮炜霖 2020101603 网络工程



20 网工　阮炜霖

10.1 (1) 直接插入
① 087, 503, 512, 061, 908, 170, 897, 275, 653, 426
② 087, 503, 512, 061, 908, 170, 897, 275, 653, 426
③ 061, 087, 503, 512, 908, 170, 897, 275, 653, 426
④ 061, 087, 503, 512, 908, 170, 897, 275, 653, 426
⑤ 061, 087, 170, 503, 512, 908, 897, 275, 653, 426
⑥ 061, 087, 170, 275, 503, 512, 897, 908, 653, 426
⑦ 061, 087, 170, 275, 503, 512, 653, 897, 908, 426
⑧ 061, 087, 170, 275, 426, 503, 512, 653, 897, 908

(2) 希尔排序
d[1]=5, d[2]=3, d[3]=1
① 170, 087, 275, 061, 426, 503, 897, 512, 653, 908
② 061, 087, 275, 170, 426, 503, 897, 512, 653, 908
③ 061, 087, 170, 275, 426, 503, 512, 653, 897, 908

(3) 快速排序
① (426, 087, 275, 061, 170), 503, (897, 908, 653, 512)
② (170, 087, 275, 061), 426, 503, (512, 653), 897, (908)
③ (061, 087), 170, (275), 426, 503, 512, (653), 897, 908
④ 061, 087, 170, 275, 426, 503, 512, 653, 897, 908

(4) 堆排　建堆 → 输出 + 逐个弹出最小值

(5) 归并：
① (087, 503), (061, 512), (170, 908), (275, 897), (426, 653)
② (061, 087, 503, 512), (170, 275, 897, 908), (426, 653)
③ (061, 087, 170, 275, 503, 512, 897, 908), (426, 653)
④ 061, 087, 170, 275, 426, 503, 512, 653, 897, 908

(6) 基数：
① ~~061, 08~~
(087, 061), (170), (275), (426), (503, 512), (653), (897), (908)
② 061, 087, 170, 275, 426, 503, 512, 653, 897, 908

10.2 Partition
① (Tim, Kay, Eva, Roy, Dot, Jon), (Kim, Ann, Tom, Jim, Guy, Amy)
② (Tim, Kay, Eva), (Roy, Dot, Jon), (Kim, Ann, Tom), (Jim, Guy, Amy)
③ (Tim, Kay), (Eva), (Roy, Dot), (Jon), (Kim, Ann), (Tom), (Jim, Guy, Amy)

Merge
① (Eva, Kay, Tim), (Dot, Jon, Roy), (Ann, Kim, Tom), (Amy, Guy, Jim)
② (Dot, Eva, Jon, Kay, Roy, Tim), (Amy, Ann, Guy, Jim, Kim, Tom)
③ (Amy, Ann, Dot, Eva, Guy, Jim, Jon, Kay, Kim, Roy, Tim, Tom)

10.7 (1) $a_n = a_{\lfloor \frac{n-1}{2} \rfloor} + a_{\lceil \frac{n-1}{2} \rceil} + n-1$ ⇒ $n=7$ 时，比较次数 $a_7 = a_3 + a_4 + 6$

理由：当两边相等或差1时最优 $= a_1 + a_2 \times 3 + 11 = 14$ 次

关键字

(2) 1, 2, 3, 4, 5, 6, 7，以4为关键字

10.11 12场

10.12 (1) 不是 100 86 48 73 35 39 42 57 66 21 是大顶堆

(2) 12 70 33 65 24 56 48 92 86 33 不是小顶堆

12 24 33 65 33 56 48 92 86 70

(3) 103 95 56 38 66 23 42 12 30 52 06 20 是大顶堆

(4) 05 56 20 23 40 38 29 61 35 76 28 100 不是

小顶堆 05 23 20 35 28 38 29 61 56 76 40 100

10.24

```cpp
//作业10.24
#include<bits/stdc++.h>
using namespace std;

void insert(int arr[], int temp[], int n){  //2-路插入排序
    int i,first,final,k;
    first=final=0;//分别记录temp数组中最大值和最小值的位置
    temp[0]=arr[0];
    for(i=1;i<n;i++){ // 待插入元素比最小的元素小
        if(arr[i]<temp[first]){
            first=(first-1+n)%n;
            temp[first] = arr[i];
        }
        else if(arr[i]>temp[final]){ //待插入元素比最大元素大
            final=(final+1+n)%n;
            temp[final]=arr[i];
        }
        else{ // 插入元素比最小大，比最大小
            k = (final + 1 + n) % n;
            while (temp[((k - 1) + n) % n] > arr[i]) { //当插入值比当前值小时，需要移动当前值的位置
                temp[(k + n) % n] =temp[(k - 1 + n) % n];
                k = (k - 1 + n) % n;
            }
            temp[(k + n) % n] = arr[i]; //插入该值
            final = (final + 1 + n) % n; //因为最大值的位置改变，所以需要实时更新final的位置
        }
    }
    for (k = 0; k < n; k ++) {    // 将排序记录复制到原来的顺序表里
        arr[k] = temp[(first + k) % n];
    }
}

signed main(){
    int a[8] = {32,1,7,75,24,45,96,61};
    int temp[8];
    insert(a,temp,8);
    for (int i = 0; i < 8; i ++){
        printf("%d ", a[i]);
    }
    return 0;
}
```

10.26

```cpp
//作业10.26
#pragma GCC optimize("Ofast", "inline", "-ffast-math")
#pragma GCC target("avx,sse2,sse3,sse4,mmx")
#include<bits/stdc++.h>
#define inf 0x3f3f3f3f
//#define int long long
using namespace std;
const int N=2e5+7;
const int mod=1e9+7;

void bubble(int a[],int tmp[],int n){
    for(int i=0;i<n;i++){
        for(int j=i+1;j<n;j++){
            if(a[j]>a[i]) swap(a[j],a[i]);
        }
    }
}

signed main(){
//  ios::sync_with_stdio(0);
//  cin.tie(0);cout.tie(0);
//  freopen("in.cpp","r",stdin);
//  freopen("out.cpp","w",stdout);
    int a[8] = {32,1,7,75,24,45,96,61};
    int temp[8];
    bubble(a,temp,8);
    for (int i = 0; i < 8; i ++){
        printf("%d ", a[i]);
    }
    return 0;
}

```