

2.1

2.1.2

选择排序算法：

首先，找到数组中最小的那个元素，其次，将它和数组的第一个元素交换位置（如果第一个元素就是最小元素那么它就和自身交换）。再次，在剩下的元素中找到最小的元素，将它与数组的第二个元素交换位置。如此往复，直到将整个数组排序。

```
public class Selection
{
    public static void sort(Comparable[] a)
    {
        // 将a[]按升序排列
        int N = a.length;           // 数组长度
        for (int i = 0; i < N; i++)
        {
            // 将a[i]和a[i+1..N]中最小的元素交换
            int min = i;             // 最小元素的索引
            for (int j = i+1; j < N; j++)
                if (less(a[j], a[min])) min = j;
            exch(a, i, min);
        }
    }
}
```

- 一个元素最多可能被交换多少次：

$N - 1$ 次。

选择排序总共需要 N 次交换，如果需要 N 次与算法思想相矛盾，不符合；如果是需要 $N - 1$ ，可以想到数组 5 1 2 3 4，5 交换了 4，即 $N - 1$ 次。

- 平均交换次数：

2。

选择排序需要 N 次交换，则有 N 次 `swap(a[i], a[j])`，则一次排序中所有元素一共交换了 $2N$ 次，一共有 N 个元素，则平均交换次数为 2。

2.1.9

希尔排序：

交换不相邻的元素以对数组的局部进行排序

```
public class Shell
{
    public static void sort(Comparable[] a)
    {
        // 将a[]按升序排列
        int N = a.length;
        int h = 1;
        while (h < N/3) h = 3*h + 1; // 1, 4, 13, 40, 121, 364, 1093, ...
        while (h >= 1)
        {
            // 将数组变为h有序
            for (int i = h; i < N; i++)
            {
                // 将a[i]插入到a[i-h], a[i-2*h], a[i-3*h]... 之中
            }
        }
    }
}
```

```

        for (int j = i; j >= h && less(a[j], a[j-h]); j -= h)
            exch(a, j, j-h);
    }
    h = h/3;
}
}
}

```

```

start: E A S Y S H E L L S O R T Q U E S T I O N
13 13 13 E A S Y S H E L L S O R T Q U E S T I O N
13 14 14 E A S Y S H E L L S O R T Q U E S T I O N
13 15 2 E A E Y S H E L L S O R T Q U S S T I O N
13 16 3 E A E S S H E L L S O R T Q U S Y T I O N
13 17 17 E A E S S H E L L S O R T Q U S Y T I O N
13 18 18 E A E S S H E L L S O R T Q U S Y T I O N
13 19 19 E A E S S H E L L S O R T Q U S Y T I O N
13 20 20 E A E S S H E L L S O R T Q U S Y T I O N
4 4 4 E A E S S H E L L S O R T Q U S Y T I O N
4 5 5 E A E S S H E L L S O R T Q U S Y T I O N
4 6 6 E A E S S H E L L S O R T Q U S Y T I O N
4 7 3 E A E L S H E S L S O R T Q U S Y T I O N
4 8 4 E A E L L H E S S S O R T Q U S Y T I O N
4 9 9 E A E L L H E S S S O R T Q U S Y T I O N
4 10 10 E A E L L H E S S S O R T Q U S Y T I O N
4 11 7 E A E L L H E R S S O S T Q U S Y T I O N
4 12 12 E A E L L H E R S S O S T Q U S Y T I O N
4 13 9 E A E L L H E R S Q O S T S U S Y T I O N
4 14 14 E A E L L H E R S Q O S T S U S Y T I O N
4 15 15 E A E L L H E R S Q O S T S U S Y T I O N
4 16 16 E A E L L H E R S Q O S T S U S Y T I O N
4 17 17 E A E L L H E R S Q O S T S U S Y T I O N
4 18 10 E A E L L H E R S Q I S T S O S Y T U O N
4 19 7 E A E L L H E O S Q I R T S O S Y T U S N
4 20 8 E A E L L H E O N Q I R S S O S T T U S Y
1 1 0 A E E L L H E O N Q I R S S O S T T U S Y
1 2 2 A E E L L H E O N Q I R S S O S T T U S Y
1 3 3 A E E L L H E O N Q I R S S O S T T U S Y
1 4 4 A E E L L H E O N Q I R S S O S T T U S Y
1 5 3 A E E H L L E O N Q I R S S O S T T U S Y
1 6 3 A E E E H L L O N Q I R S S O S T T U S Y
1 7 7 A E E E H L L O N Q I R S S O S T T U S Y
1 8 7 A E E E H L L N O Q I R S S O S T T U S Y
1 9 9 A E E E H L L N O Q I R S S O S T T U S Y
1 10 5 A E E E H I L L N O Q R S S O S T T U S Y
1 11 11 A E E E H I L L N O Q R S S O S T T U S Y
1 12 12 A E E E H I L L N O Q R S S O S T T U S Y
1 13 13 A E E E H I L L N O Q R S S O S T T U S Y
1 14 10 A E E E H I L L N O O Q R S S S T T U S Y
1 15 15 A E E E H I L L N O O Q R S S S T T U S Y
1 16 16 A E E E H I L L N O O Q R S S S T T U S Y
1 17 17 A E E E H I L L N O O Q R S S S T T U S Y
1 18 18 A E E E H I L L N O O Q R S S S T T U S Y
1 19 16 A E E E H I L L N O O Q R S S S S T T U Y
1 20 20 A E E E H I L L N O O Q R S S S S T T U Y
end: A E E E H I L L N O O Q R S S S S T T U Y

```

```

void solve() {
    string s = "EASYSHELLSORTQUESTION";
    int n = s.size();
    vector<char> a(n);
    for(int i = 0; i < n; ++i) a[i] = s[i];
    auto print = [&]() -> void {
        for(auto &t: a) cout<<t<<" "; puts("");
    };
    auto ssort = [&]() -> void {
        int N = a.size();
        int h = 1;
        while(h < N / 3) h = 3 * h + 1;
        cout<<"start: "; print();
        while(h >= 1) {
            for(int i = h; i < N; ++i) {
                int j;
                for(j = i; j >= h && a[j] < a[j - h]; j-= h) swap(a[j], a[j -
h]);
                printf("%3d %3d %3d ", h,i,j); print();
            }
            h = h / 3;
        }
        cout<<"end: "; print();
    };
    ssort();
}

```

2.2

2.2.2

```

merge( 0, 0, 1): A E S Y Q U E S T I O N
merge( 0, 1, 2): A E S Y Q U E S T I O N
merge( 3, 3, 4): A E S Q Y U E S T I O N
merge( 3, 4, 5): A E S Q U Y E S T I O N
merge( 0, 2, 5): A E Q S U Y E S T I O N
merge( 6, 6, 7): A E Q S U Y E S T I O N
merge( 6, 7, 8): A E Q S U Y E S T I O N
merge( 9, 9, 10): A E Q S U Y E S T I O N
merge( 9, 10, 11): A E Q S U Y E S T I N O
merge( 6, 8, 11): A E Q S U Y E I N O S T
merge( 0, 5, 11): A E E I N O Q S S T U Y
A E E I N O Q S S T U Y

```

```

template<typename T>
class TopMerge {
private:
    vector<T> vec,tmp;
public:
    TopMerge() {}
    TopMerge(const typename vector<T>::iterator& beg,const typename
vector<T>::iterator& end) {
        vec.assign(beg, end);
    }

```

```

        tmp.assign(beg, end);
        msort(0, vec.size()-1);
        info();
    }
    void msort(int lo, int hi) {
        if(lo >= hi) return ;
        int mid = hi + lo >> 1;
        msort(lo, mid), msort(mid+1, hi);
        merge(lo, mid, hi);
        print(lo, mid, hi);
    }
    void merge(int lo, int mid, int hi) {
        int i = lo, j = mid + 1, k = 0;
        while(i <= mid && j <= hi) {
            if(vec[i] > vec[j]) tmp[k++] = vec[j++];
            else tmp[k++] = vec[i++];
        }
        while(i <= mid) tmp[k++] = vec[i++];
        while(j <= hi) tmp[k++] = vec[j++];

        for(i = lo, j = 0; i <= hi; ++i, ++j) vec[i] = tmp[j];
    }
    void info() {
        for(auto &t: vec) cout<<t<<" "; puts("");
    }
    void print(int lo, int mid, int hi) {
        printf("merge( %2d, %2d, %2d): ", lo, mid, hi);
        for(auto &t: vec) cout<<t<<" "; puts("");
    }
};

```

2.3

2.3.1

初始值: 0 11 E A S Y Q U E S T I O N
 扫描左右部分: 2 6 E A S Y Q U E S T I O N
 交换: 2 6
 最后一次交换: 3 2 E A E Y Q U S S T I O N
 结果: 2
 最后结果: 2 E A E Y Q U S S T I O N

```

template<typename T>
class QSort {
private:
    vector<T> vec;
public:
    QSort() {}
    QSort(const typename vector<T>::iterator& beg, const typename
vector<T>::iterator& end) {
        vec.assign(beg, end);
        qsort(0, vec.size()-1);
    }
    void qsort(int lo, int hi) {
        if(lo >= hi) return ;
    }
};

```

```

        cout<<"初始值: "<<lo<<" "<<hi<<" "; info();
        int partIdx = partition(lo,hi);
        cout<<"最后结果: "<<partIdx<<" "; info();
        return ;

        qsort(lo, partIdx-1), qsort(partIdx+1, hi);
        return ;
    }
    int partition(int lo, int hi) {
        int i = lo, j = hi + 1;
        T tmp = vec[lo];
        while(true) {
            while(vec[++i] < tmp) if(i == hi) break;
            while(tmp < vec[--j]) if(j == lo) break;
            if(i >= j) break;
            cout<<"扫描左右部分: "<<i<<" "<<j<<" "; info();
            cout<<"交换: "<<i<<" "<<j<<endl;
            swap(vec[i], vec[j]);
        }
        cout<<"最后一次交换: "<<i<<" "<<j<<" "; info();
        cout<<"结果: "<<j<<endl;
        swap(vec[lo], vec[j]);
        return j;
    }
    void info() {
        for(auto &t: vec) cout<<t<<" "; puts("");
    }
};

```

2.3.2

```

(lo, j, ho) ==> ( 0,  2, 11) E A E Y Q U S S T I O N
(lo, j, ho) ==> ( 0,  1,  1) A E E Y Q U S S T I O N
(lo, j, ho) ==> ( 3, 11, 11) A E E N Q U S S T I O Y
(lo, j, ho) ==> ( 3,  4, 10) A E E I N U S S T Q O Y
(lo, j, ho) ==> ( 5, 10, 10) A E E I N O S S T Q U Y
(lo, j, ho) ==> ( 5,  5,  9) A E E I N O S S T Q U Y
(lo, j, ho) ==> ( 6,  7,  9) A E E I N O Q S T S U Y
(lo, j, ho) ==> ( 8,  9,  9) A E E I N O Q S S T U Y

```

```

template<typename T>
class QSort {
private:
    vector<T> vec;
public:
    QSort() {}
    QSort(const typename vector<T>::iterator& beg, const typename
vector<T>::iterator& end) {
        vec.assign(beg, end);
        qsort(0, vec.size()-1);
    }
    void qsort(int lo, int hi) {
        if(lo >= hi) return ;
        int partIdx = partition(lo,hi);
        print(lo,partIdx,hi);
        qsort(lo, partIdx-1), qsort(partIdx+1, hi);
    }
};

```

```

        return ;
    }
    int partition(int lo, int hi) {
        int i = lo, j = hi + 1;
        T tmp = vec[lo];
        while(true) {
            while(vec[++i] < tmp) if(i == hi) break;
            while(tmp < vec[--j]) if(j == lo) break;
            if(i >= j) break;
            swap(vec[i], vec[j]);
        }
        swap(vec[lo], vec[j]);
        return j;
    }
    void info() {
        for(auto &t: vec) cout<<t<<" ";puts("");
    }
    void print(int lo, int ptidx, int ho) {
        printf("(lo, j, ho) ==> (%2d, %2d, %2d) ",lo, ptidx, ho); info();
    }
};

```

2.4

2.4.6

第 1次

P

第 2次

R -> P

第 3次

R -> P

R -> I

第 4次

R -> P

P -> O

R -> I

第 5次

Pop: R

P -> O

P -> I

第 6次

R -> P

P -> O

R -> I

第 7次

Pop: R

P -> O

P -> I

第 8次

Pop: P

O -> I

第 9次

O -> I

O -> I

第 10次

Pop: O

I -> I

第 11次

T -> I

T -> I

第 12次

Pop: T

I -> I

第 13次

Y -> I

Y -> I

第 14次

Pop: Y

I -> I

第 15次

Pop: I

I

第 16次

Pop: I

第 17次

Q

第 18次

U -> Q

第 19次

U -> Q

U -> E

第 20次

Pop: U

Q -> E

第 21次

Pop: Q

E

第 22次

Pop: E

第 23次

U

第 24次

Pop: U

第 25次

E

```
template<typename T>
class PriorityQueue {
private:
    const int MaxN = 1e2 + 21;
    vector<T> vec;
    bitset<1030> vis;
    int N = 0;
public:
    PriorityQueue() {
        vec.assign(MaxN, T{});
    }
    void swim(int k) {
        while(k > 1 && vec[k / 2] < vec[k]) {
            swap(vec[k / 2], vec[k]);
            k >>= 1;
        }
    }
    void sink(int k) {
        while(2 * k <= N) {
            int j = 2 * k;
            if(j < N && vec[j] < vec[j + 1]) ++j;
            if(vec[k] >= vec[j]) break;
            swap(vec[k], vec[j]);
            k = j;
        }
    }
    void insert(T v) {
        vec[++N] = v;
        vis[N] = 1;
        swim(N);
    }
    T pop(){
        T ma = vec[1];
        vis[N] = 0;
        swap(vec[1], vec[N--]);
        sink(1);
        return ma;
    }
    void print() {
        if(N == 1) {
            cout<<vec[1]<<endl;
            return ;
        }
        dfs(1);
    }
    void dfs(int root) {
        int lson = root << 1;
        if(vis[lson]) {
            cout<<vec[root]<<" -> "<< vec[lson]<<endl;
            dfs(lson);
        }
    }
};
```



```

    }
    if(vis[lson + 1]) {
        cout<<vec[root]<<" -> "<<vec[lson+1]<<endl;
        dfs(lson + 1);
    }
}

};

void solve() {
    string s = "PRIO*R**I*T*Y***QUE***U*E";
    PriorityQueue<char> heap;
    for(int i = 0; i < s.size(); ++i) {
        cout<<"第 "<<i+1<<"次"<<endl;
        if(s[i] == '*') {
            cout<<"Pop: "<<heap.pop()<<endl;
            heap.print();
        } else {
            heap.insert(s[i]);
            heap.print();
        }
        cout<<endl;
    }
}

```

2.4.9

A B C D E

排列，一个一个找。

```

8
1
E -> D
D -> A
D -> C
E -> B

2
E -> D
D -> A
D -> B
E -> C

3
E -> C
C -> A
C -> B
E -> D

4
E -> D
D -> B
D -> C
E -> A

5
E -> D

```

D -> B
D -> A
E -> C

6
E -> C
C -> B
C -> A
E -> D

7
E -> D
D -> C
D -> B
E -> A

8
E -> D
D -> C
D -> A
E -> B

```
class strHash {
    typedef unsigned long long ULL;
public:
    strHash(const string& s) {
        this->str = "^" + s;
        dispose();
    }
    ULL get(int l, int r) {
        return h[r] - h[l - 1] * p[r - l + 1];
    }
private:
    void dispose() {
        len = str.size();
        h.assign(len + 1, 0); p.assign(len + 1, 0);
        h[0] = p[0] = 1;
        for(int i = 1; i <= len; ++i) {
            h[i] = h[i-1] * P + str[i];
            p[i] = p[i - 1] * P;
        }
    }

    const ULL P = 11451;
    string str;
    ULL len;
    vector<ULL> h,p;
};

template<typename T>
class PriorityQueue {
private:
    const int MaxN = 1e2 + 21;
    bitset<1030> vis;
```

```

public:
    vector<T> vec;
    int N = 0;
    PriorityQueue() {
        vec.assign(MaxN, T{});
    }
    void swim(int k) {
        while(k > 1 && vec[k / 2] < vec[k]) {
            swap(vec[k / 2], vec[k]);
            k >>= 1;
        }
    }
    void sink(int k) {
        while(2 * k <= N) {
            int j = 2 * k;
            if(j < N && vec[j] < vec[j + 1]) ++j;
            if(vec[k] >= vec[j]) break;
            swap(vec[k], vec[j]);
            k = j;
        }
    }
    void insert(T v) {
        vec[++N] = v;
        vis[N] = 1;
        swim(N);
    }
    T pop(){
        T ma = vec[1];
        vis[N] = 0;
        swap(vec[1], vec[N--]);
        sink(1);
        return ma;
    }
    void print() {
        if(N == 1) {
            cout<<vec[1]<<endl;
            return ;
        }
        dfs(1);
    }
    void dfs(int root) {
        int lson = root << 1;
        if(vis[lson]) {
            cout<<vec[root]<<" -> "<< vec[lson]<<endl;
            dfs(lson);
        }
        if(vis[lson + 1]) {
            cout<<vec[root]<<" -> "<<vec[lson+1]<<endl;
            dfs(lson + 1);
        }
    }
};

void solve() {
    string s = "ABCDE";
    vector<PriorityQueue<char>> vec;
    typedef unsigned long long ULL;
    set<ULL> st;

```

```

do {
    PriorityQueue<char> q;
    for(auto &t: s) q.insert(t);
    string tmp = "";
    for(int i = 1; i <= q.N; ++i) tmp += q.vec[i];
    strHash hash(tmp);
    if(!st.count(hash.get(1, tmp.size()))) {
        st.insert(hash.get(1, tmp.size()));
        vec.push_back(q);
    }

} while(next_permutation(all(s)));

cout<<st.size()<<endl;
for(int i = 0; i < vec.size(); ++i) {
    cout<<i + 1<<endl;
    vec[i].print();
    puts("");
}
}

```

A A A B B

同理

```

2
1
B(1) -> B(2)
B(2) -> A(4)
B(2) -> A(5)
B(1) -> A(3)

2
B(1) -> A(2)
A(2) -> A(4)
A(2) -> A(5)
B(1) -> B(3)

```