

基础规则

- 无需给出具体细节

如swap

- 用-->赋值，更标准

循环

缩进

除了函数无冒号:

downto

if 带括号

for 带括号，带do

```
1  for (i ← 1 to n - 1) do
2      for (j ← n downto i + 1) do
3          if (A[j] < A[j - 1]) then
4              Swap A[j] and A[j - 1]
```

数据结构

数组

- 统计数组长度

```
1  A.length()
```

- 传递&定义数组：要给下标区间

```
1  left[1,n/2]=Mergesort(A[1,n/2])
2  right[1,n/2]=Mergesort(A[n/2+1,n])
3  sol[1,n]= Merge(left[1,n/2],right[1,n/2])
```

优先队列

```
1 Q.ExtractMin()
2 Q.update(v,v.dist)
```

堆&栈

```
1 S.push(x)//insert(x)
2 S.pop()//extractmax(A)
3 S.empty()
4 S.top()//A[1]
```

LIFO

```
1 Deque Q//需要初始化
2 Q.addlast(A[i])
3 Q.addlast(Merge(L,R))
4 p=Q.removefirst()
```

list&桶

```
1 L.front//访问首元素
2 L.pop_front()//踢出首元素
3 L.push_back(x)//从尾部推入
4 L.empty()
5 concatenate(l1,l2,l3)//连接list
6 concat(B1,B2,B3)//连接桶
```

二元对

跟踪记录很方便

```
1 cp=make_pair(a,b)
2 a=cp.first//访问二元对
3 b=cp.second
```

规律总结

需初始化

```
1 Deque Q//LIFO
2 list L//列表
3 min_heap q
4 Build a priority queue Q based on 'dist' values//建立了二元对
5
```

类自然语言

拓扑排序

```
1 Run DFS to obtain topological order
2 for(each node u in topological order)
```

随机选择

```
1 Pick an arbitrary node x
```