



链滴

数据结构与算法 (动画) 安卓 app

作者: [Wjwy](#)

原文链接: <https://ld246.com/article/1593499692282>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

下载地址(腾讯云对象存储服务):

1.算法动画图解

[点击此处直接下载](#)

<https://hgxs-1253461189.cos.ap-guangzhou.myqcloud.com/sfdhtj.apk>

2.数据结构与算法教程

[点击此处直接下载](#)

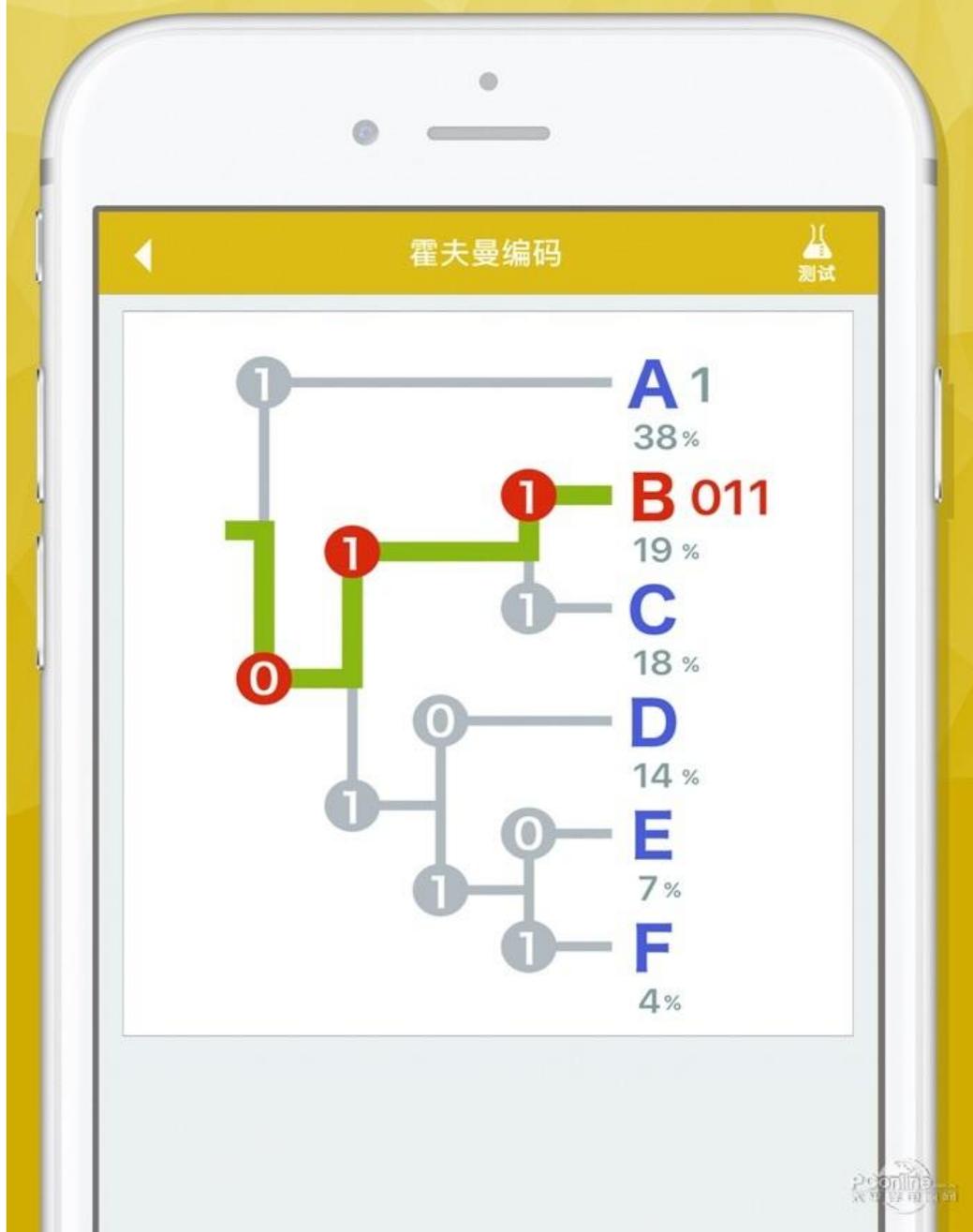
<https://hgxs-1253461189.cos.ap-guangzhou.myqcloud.com/sjjgysf.apk>

算法动画图解

享受观看，尝试和学习的算法动画图解。算法的广泛领域用动画清晰简洁地解释。可以各种尝试的“试模式”来加深您的理解。还充满了您熟悉地加密和安全等算法。来吧，让我们进入算法的世界！

通过“测试模式” 你可以试试它

通过在各种条件下尝试移动来加深对算法的理解。



通过“学习模式” 观看和学习

动画和全面的解释有助于你理解算法
的工作原理。



从候选点中选择成本最低的点。在这种情况下，它是点D。

算法彻底解释

分类、聚类、列表搜索、图表搜索、数学、
数据压缩、安全、数据结构、网络、递归



算法动画图解

分类

📖 冒泡排序 ▶

🧪 测试 ▶

📖 选择性排序 ▶

🧪 测试 ▶

📖 插入排序 ▶

🧪 测试 ▶



充满了您想 知道的熟悉地算法

详细解释与数据压缩，加密和安全技术等
IT相关的概念。



==== 类别和包含的主题 =====

[分类]

冒泡排序、选择性排序、插入排序、堆排序、归并排序、快速排序

[聚类]

K-均值算法

[列表搜索]

线性搜索、二分搜索

[图表搜索]

广度优先搜索、深度优先搜索、贝尔曼-福特算法、戴克斯特拉算法、A*搜索算法

[数学]

辗转相除法、素性测试

[数据压缩]

游程编码、霍夫曼编码

[安全]

哈希函数、共有密钥加密、公开密钥加密、混合加密、迪菲-赫尔曼密钥交换、讯息鉴别码、数字签、数字证书

[数据结构]

列表、数组、栈、队列、哈希表、堆、二叉查找树

[网络]

佩奇排名（网页排名）

[递归]

汉诺塔

1.下载地址

<https://hgxs-1253461189.cos.ap-guangzhou.myqcloud.com/sfdhtj.apk>

数据结构与算法教程

这是一款数据结构与算法的辅助理解工具，提供了丰富的动画演示、模拟场景来帮助你更好的理解抽象的数据结构和复杂的算法。

通过自己创建的模型，你可以更深入的理解算法的细节和数据结构的特性。

是你学习和面试的好帮手。



8:39



表

线性表



链表



双向链表



多项式相加

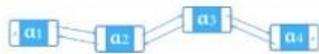


多项式相加模拟器



双向链表

双向链表和单链表很类似，唯一的区别是每个节点都有对列表中下一个节点和前一个节点的引用。



双向链表的插入

假设要插入的节点为s，要将s接入到p和q之间，只需要：

```
s->prior=p;  
s->next=p->next;  
p->next->prior=s;  
p->next=s;
```



双向链表的删除

假设要删除的节点为p，只需要：

```
p->prior->next = p->next;  
p->next->prior=p->prior
```



双向链表

双向链表的删除

C++ Java

双向链表和单链表很类似，唯一的区别是每个节点都有对列表中下一个节点和前一个节点的引用。

```

template <typename T>
void DoublyLinkedList<T>::remove(int index)
{
    if (index < 0) {
        cout << "index < 0 " << endl;
        return;
    }

```

双向链表的插入

假设要插入的节点为s，要将s接入到p和q之间，只需要：

```

int temp = 0;
Node<T>* pNode = header;
while (temp < index) {
    pNode = pNode->next;
    temp++;
}

```



```

if (pNode == nullptr) {
    cout << "index out of bounds! " << endl;
    return;
}
pNode->prev->next = pNode->next;
if (pNode->next != nullptr) {
    pNode->next->prev = pNode->prev;
}

```

双向链表的删除

假设要删除的节点为p，只需要：

```

p->prior->next = p->next;
p->next->prior = p->prior

```



播放 代码



8:43

数据结构与算法



表



栈&队列



树



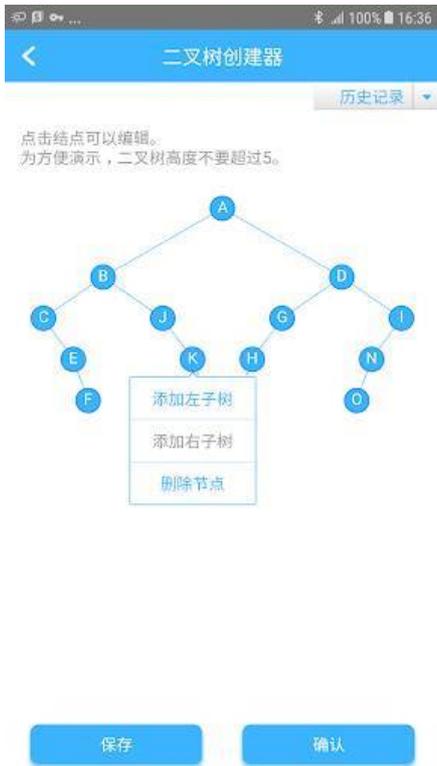
查找



排序



设置



2. 下载地址

<https://hgxs-1253461189.cos.ap-guangzhou.myqcloud.com/sjjgysf.apk>