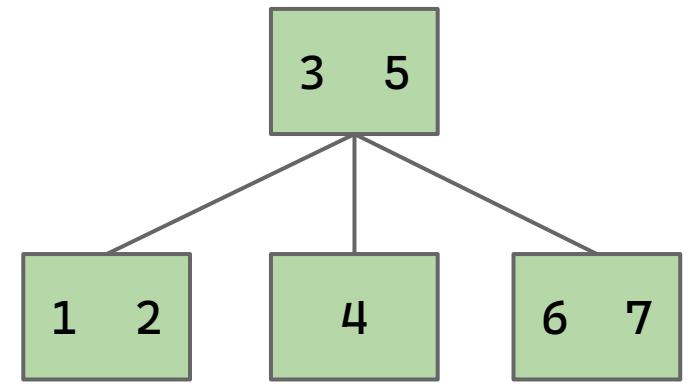


BSTs, B-trees, AVL trees, Red-black trees

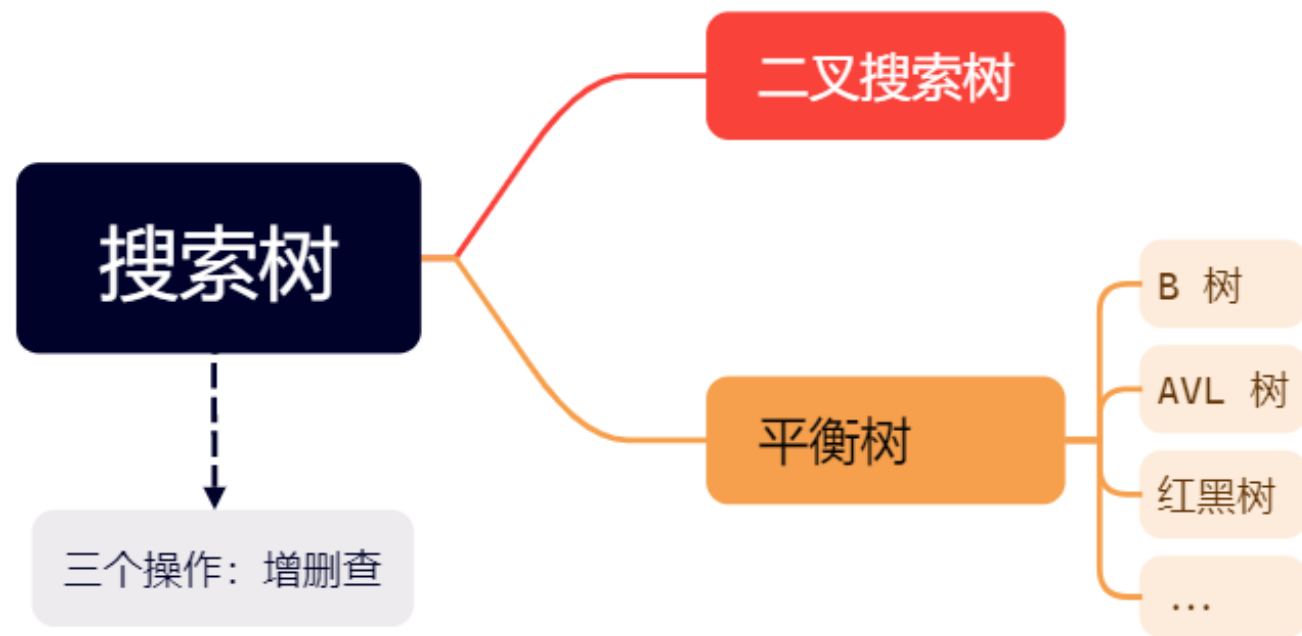
主讲人: 七海Nana7mi
课程大纲: CS61B



课程说明：

- 课程内容基于 UC-Berkeley 的课程 **CS61B-sp18** 与 **CS61B-fa23**。可以理解为课程的汉化视频。
- 课程使用的编程语言为 **Java**。
- **AI** 语音模型来源 **BiliBili** 用户 **Xz乔希**。
- 七海也在学习中，有错误敬请指出！

章节目录



二叉搜索树： 导入

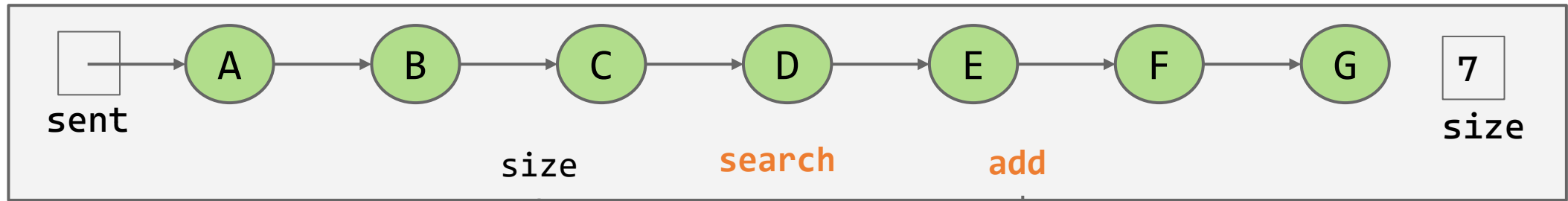
Lecture 1

二叉搜索树

- 导入
- 定义
- ``contains()'`
- ``insert()'`
- ``delete()'`

二叉搜索树的应用

For the *order linked list* implementation below,
an operation of search can take worst case
linear time, $\theta(N)$.



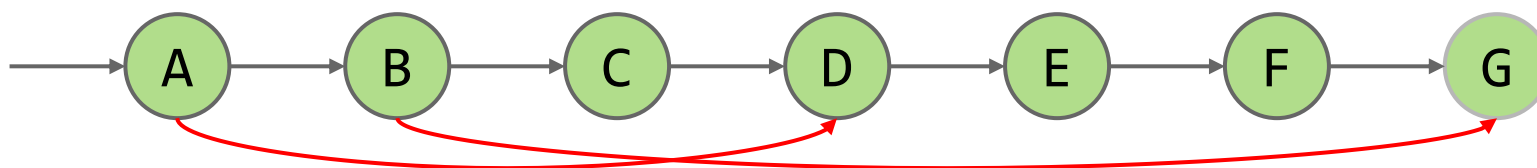
时间复杂度: $O(N)$

- **How to do?**

Fundamental Problem:

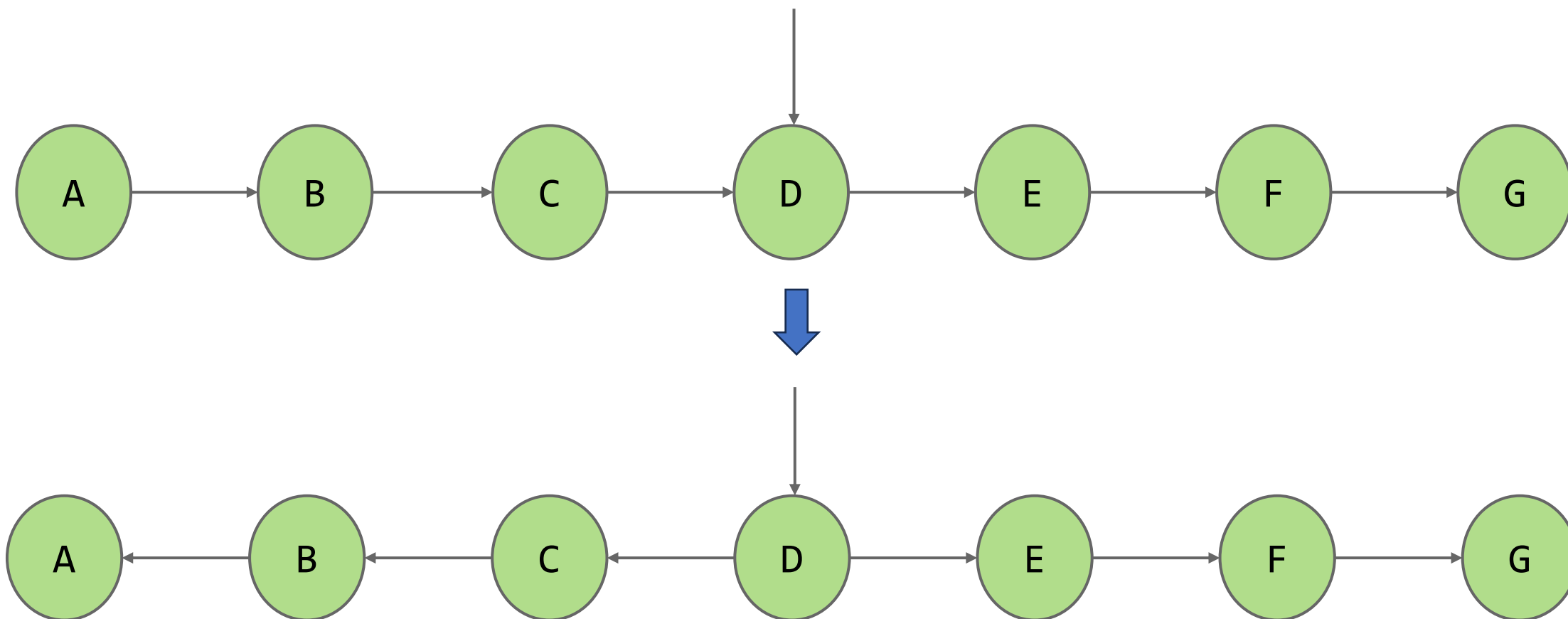
Slow search, even though it's in order.

- 我们可以任意的增加不同元素之间的连接线，来缩短路程



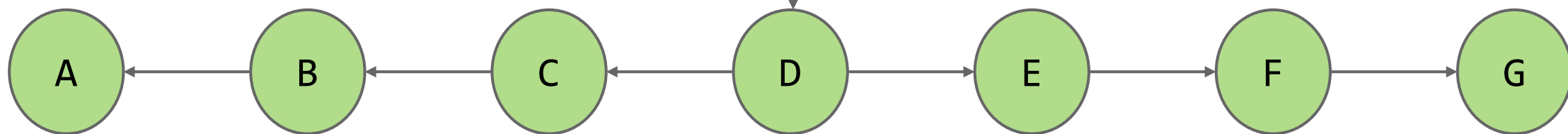
这种方法可以改进为一中数据结构也被叫做跳跃列表，我们对他的讲解讲止步于此，感兴趣的脆鲨可以自行搜索。

- 利用有序性，我们可以把入口指针指向中间的元素



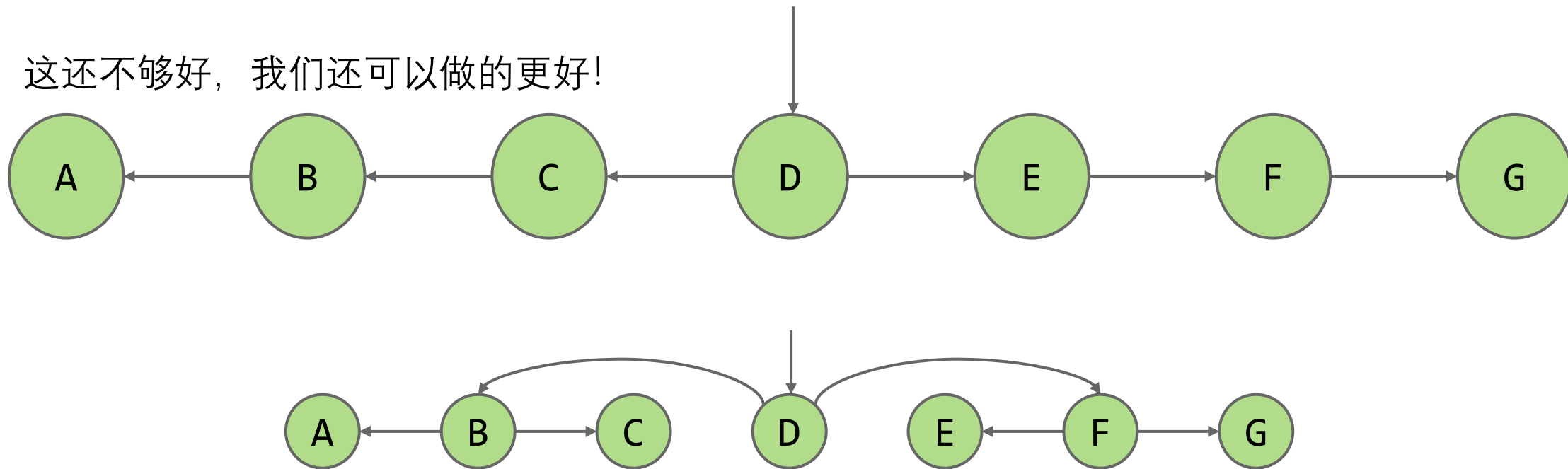
- 利用有序性，我们可以把入口指针指向中间的元素

这还不够好，我们还可以做的更好！



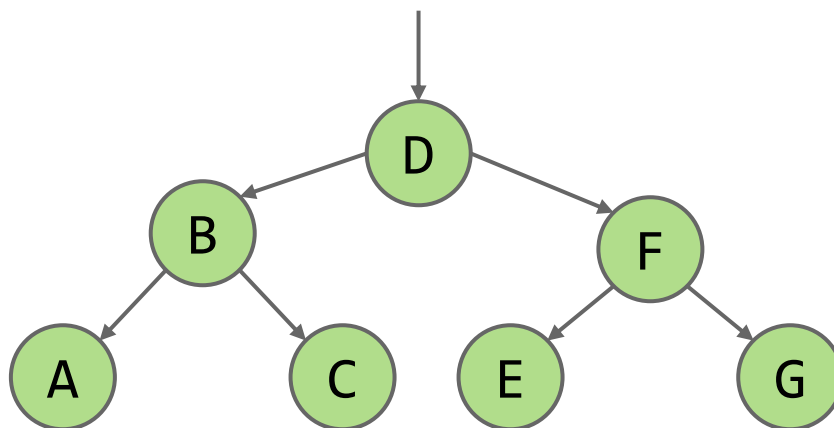
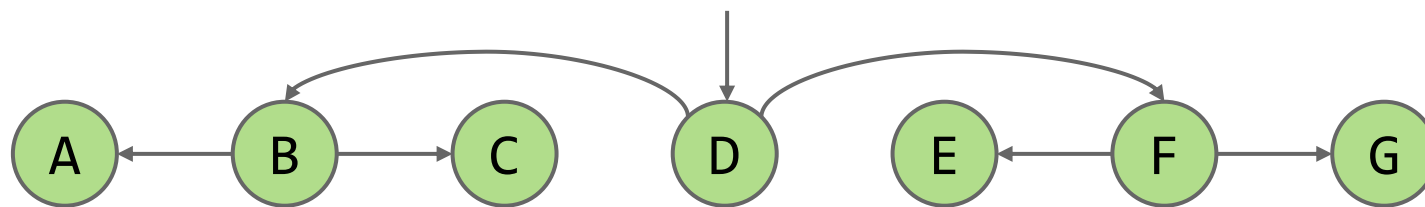
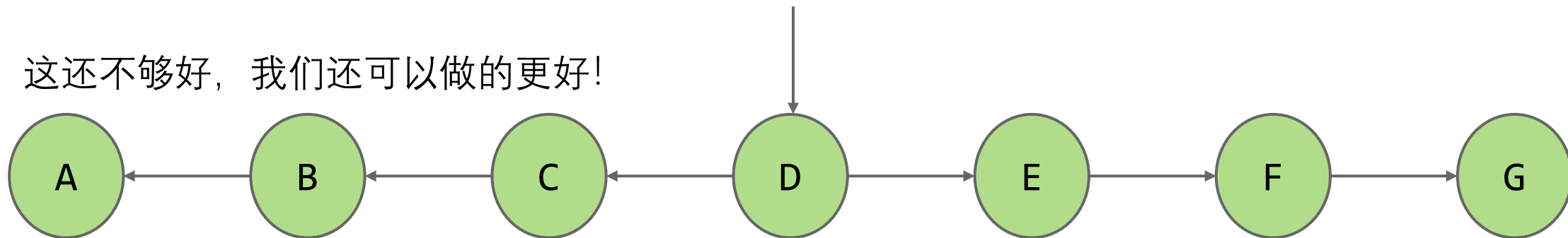
- 利用有序性，我们可以把入口指针指向中间的元素

这还不够好，我们还可以做的更好！



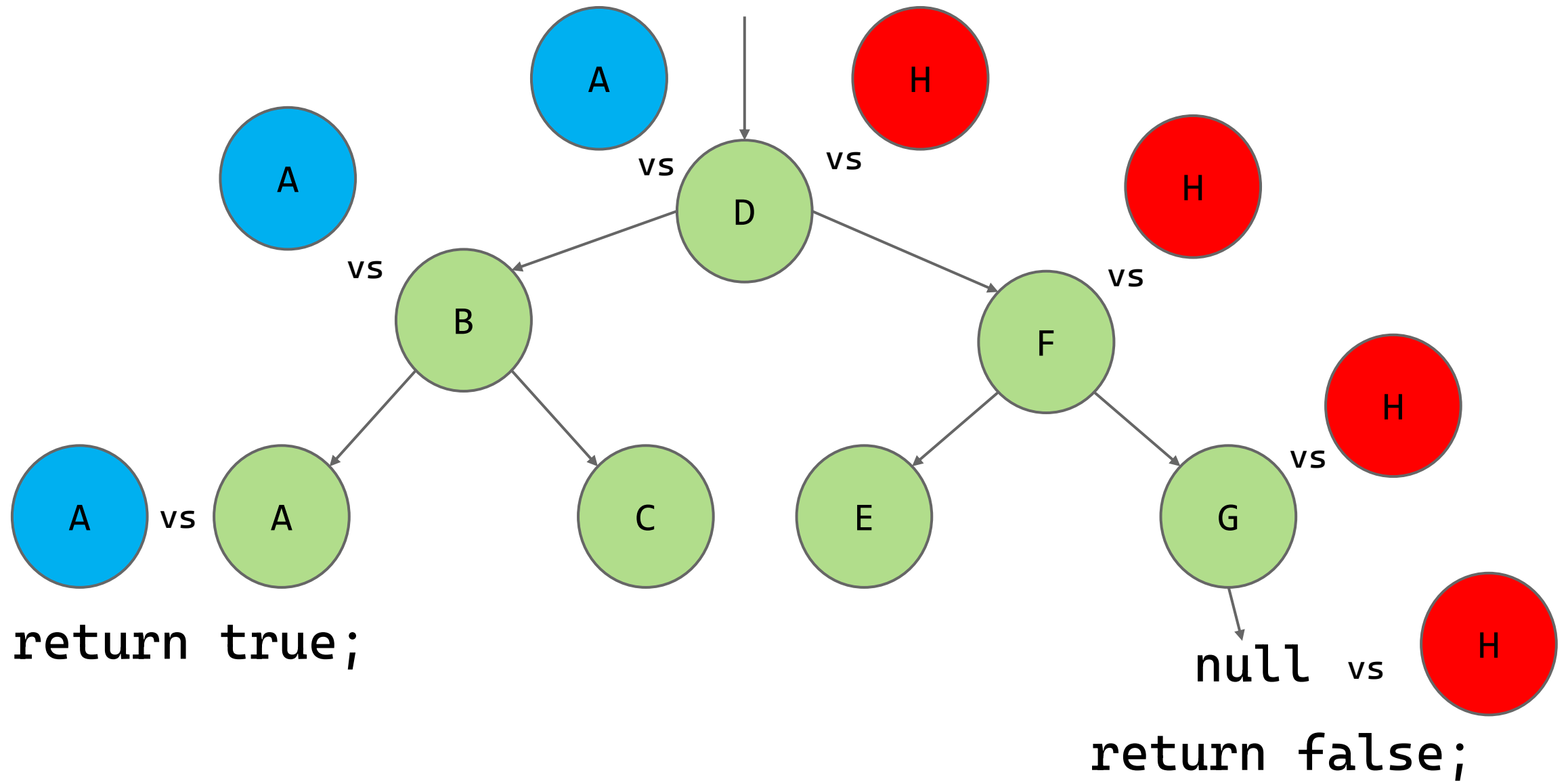
- 利用有序性，我们可以把入口指针指向中间的元素

这还不够好，我们还可以做的更好！



contains('A')?

contains('H')?



二叉搜索树： 定义

Lecture 1

二叉搜索树

- 导入
- 定义
- `contains()`
- `insert()`
- `delete()`

二叉搜索树的应用

二叉搜索树： `contains()`

Lecture 1

二叉搜索树

- 导入
- 定义
- `contains()`
- `insert()`
- `delete()`

二叉搜索树的应用

二叉搜索树： `insert()`

Lecture 1

二叉搜索树

- 导入
- 定义
- `contains()`
- **`insert()`**
- `delete()`

二叉搜索树的应用

二叉搜索树： delete()

Lecture 1

二叉搜索树

- 导入
- 定义
- contains()
- insert()
- **delete()**

二叉搜索树的应用

二叉搜索树： 应用

Lecture 1

二叉搜索树

- 导入
- 定义
- `contains()`
- `insert()`
- `delete()`

二叉搜索树的应用

