3.2 Binary Search Tree Demo

click to begin demo
Binary search tree operations

**Search.** If less, go left; if greater, go right; if equal, search hit.

successful search for H
Binary search tree operations

**Search.** If less, go left; if greater, go right; if equal, search hit.

---

**successful search for H**

---

black nodes could match the search key

---

compare H and S (go left)

---
Search. If less, go left; if greater, go right; if equal, search hit.
**Binary search tree operations**

**Search.** If less, go left; if greater, go right; if equal, search hit.

---

**successful search for H**

compare H and E  
(go right)

---

The diagram shows a binary search tree with nodes A, C, H, M, R, E, and S. The tree structure is as follows:

- Root: S
- Left subtree of S: E
  - Left child of E: H
    - Left child of H: C
    - Right child of H: M
  - Right child of E: R
- Right subtree of S: X

The search for H starts at the root (S) and proceeds by comparing the search value (H) with the node value (E). Since H is greater than E, the search moves to the right child of E, which is H. Finally, the search is successful.
**Binary search tree operations**

**Search.** If less, go left; if greater, go right; if equal, search hit.

**successful search for H**
Binary search tree operations

**Search.** If less, go left; if greater, go right; if equal, search hit.

successful search for H

![Binary search tree diagram](image)
Binary search tree operations

Search. If less, go left; if greater, go right; if equal, search hit.

successful search for H
Binary search tree operations

Search. If less, go left; if greater, go right; if equal, search hit.

successful search for H
Binary search tree operations

Search. If less, go left; if greater, go right; if equal, search hit.

unsuccessful search for G
Binary search tree operations

**Search.** If less, go left; if greater, go right; if equal, search hit.

**unsuccessful search for G**

compare G and S (go left)
Binary search tree operations

**Search.** If less, go left; if greater, go right; if equal, search hit.

**unsuccessful search for G**
**Binary search tree operations**

**Search.** If less, go left; if greater, go right; if equal, search hit.

**unsuccessful search for G**

compare G and E (go right)

![Diagram of binary search tree operations](image)
**Search.** If less, go left; if greater, go right; if equal, search hit.

unsuccessful search for G
Binary search tree operations

**Search.** If less, go left; if greater, go right; if equal, search hit.

unsuccessful search for G
Binary search tree operations

**Search.** If less, go left; if greater, go right; if equal, search hit.

unsuccessful search for G
Binary search tree operations

**Search.** If less, go left; if greater, go right; if equal, search hit.

unsuccessful search for G

compare G and H (go left)
Binary search tree operations

**Search.** If less, go left; if greater, go right; if equal, search hit.

Unsuccessful search for G
Binary search tree operations

**Search.** If less, go left; if greater, go right; if equal, search hit.

unsuccessful search for G

no more tree (search miss)
**Binary search tree operations**

**Insert.** If less, go left; if greater, go right; if null, insert.

**insert G**
**Binary search tree operations**

**Insert.** If less, go left; if greater, go right; if null, insert.

![Binary search tree diagram]

- **insert G**
  - Compare G and S (go left)

```plaintext
E  S
  /   \
A    R
 /     \
C     H
     /   \
    M    
```
**Binary search tree operations**

**Insert.** If less, go left; if greater, go right; if null, insert.

---

**Diagram:**

```
  S
  /  
X   
  
  G
  /  
E  
  /  
A  
  /  
C  
  /  
H  
  /  
M  
```

**Note:**
- **insert G**
**Binary search tree operations**

**Insert.** If less, go left; if greater, go right; if null, insert.

**insert G**

compare G and E (go right)

![Binary search tree diagram](image-url)
Insert. If less, go left; if greater, go right; if null, insert.

insert G
Binary search tree operations

**Insert.** If less, go left; if greater, go right; if null, insert.

insert G

![Binary search tree diagram](image-url)
Binary search tree operations

**Insert.** If less, go left; if greater, go right; if null, insert.

-insert G-
Binary search tree operations

Insert. If less, go left; if greater, go right; if null, insert.

insert G
**Binary search tree operations**

**Insert.** If less, go left; if greater, go right; if null, insert.

```
insert G
```
**Binary search tree operations**

**Insert.** If less, go left; if greater, go right; if null, insert.

```plaintext
insert G
```

![Binary search tree diagram](image)
Insert. If less, go left; if greater, go right; if null, insert.

insert G
Binary search tree operations

**Insert.** If less, go left; if greater, go right; if null, insert.

-insert G-