**Lecture 8: Augment**

**OS-Select(x, i)**

**{**

**DFS\_Visit(u)**

**{**

**u->color = GREY;**

**time = time+1;**

**u->d = time;**

**for each v ∈ u->Adj[]**

**{**

**if (v->color == WHITE)**

**DFS\_Visit(v);**

**}**

**u->color = BLACK;**

**time = time+1;**

**u->f = time;**

**}**

**r = x->left->size + 1;**

**MST-Prim(G, w, r)**

**Q = V[G];**

**for each *u* ∈ Q**

**key[u] = ∞;**

**key[r] = 0;**

**p[r] = NULL;**

**while (Q not empty)**

**u = ExtractMin(Q);**

**for each *v* ∈ Adj[*u*]**

**if (v ∈ Q and w(*u,v*) < key[*v*])**

**p[v] = u;**

**key[v] = w(u,v);**

**if (i == r)**

**return x;**

**else if (i < r)**

**return OS-Select(x->left, i);**

**else**

**return OS-Select(x->right, i-r);**

**}**

**IntervalSearch(T, i)**

**{**

**x = T->root;**

**while (x != NULL && !overlap(i, x->interval))**

**if (x->left != NULL && x->left->max ≥ i->low)**

**x = x->left;**

**else**

**x = x->right;**

**return x**

**}**

**Lecture 9: BFS**

**DFS(G)**

**{**

**for each vertex u ∈ G->V**

**{**

**u->color = WHITE;**

**}**

**time = 0;**

**for each vertex u ∈ G->V**

**{**

**if (u->color == WHITE)**

**DFS\_Visit(u);**

**}**

**}**

**BFS(G, s) {**

**initialize vertices;**

**Q = {s}; *// Q is a queue (duh); initialize to s***

**while (Q not empty) {**

**u = RemoveTop(Q);**

**for each v ∈ u->adj {**

**if (v->color == WHITE)**

**v->color = GREY;**

**v->d = u->d + 1;**

**v->p = u;**

**Enqueue(Q, v);**

**}**

**u->color = BLACK;**

**}**

**}**

|  |  |  |
| --- | --- | --- |
|  | Adj Matrix | Adj List |
| test (u, v) ∈ E | Θ(1) | O(n) |
| Degree(u) | Θ(n) | O(n) |
| Memory | Θ(n2) | Θ(n+m) |
| Edge insertion | Θ(1) | Θ(1) |
| Edge deletion | Θ(1) | O(n) |
| Graph traversal | Θ(n2) | Θ(n+m) |

**Final Review Slides**

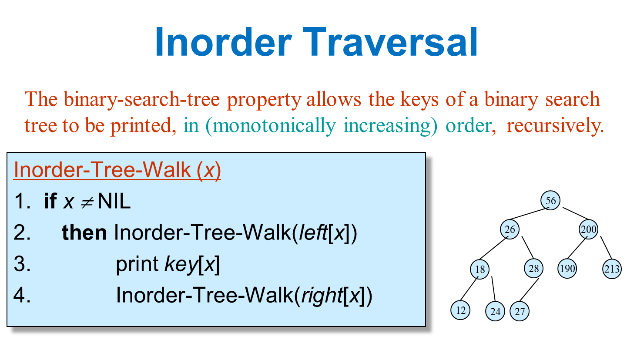
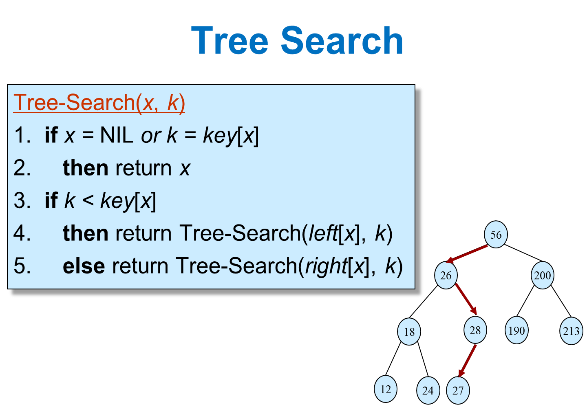
Graphical user interface, text

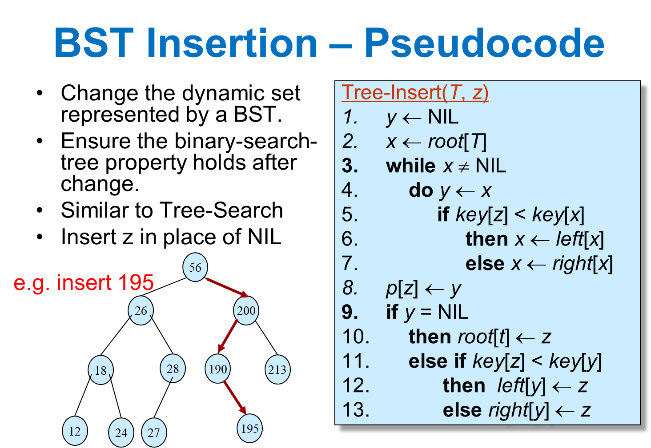
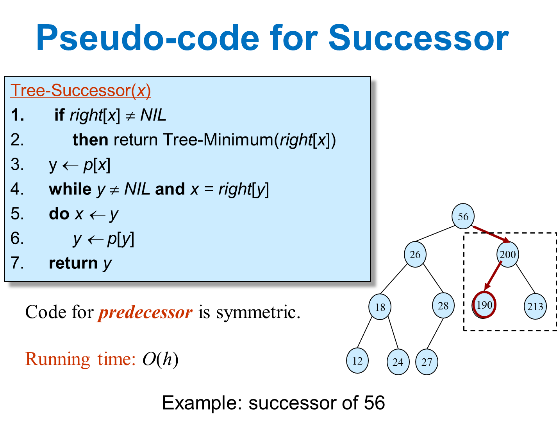
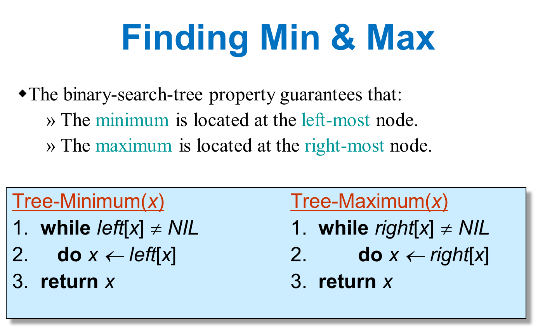
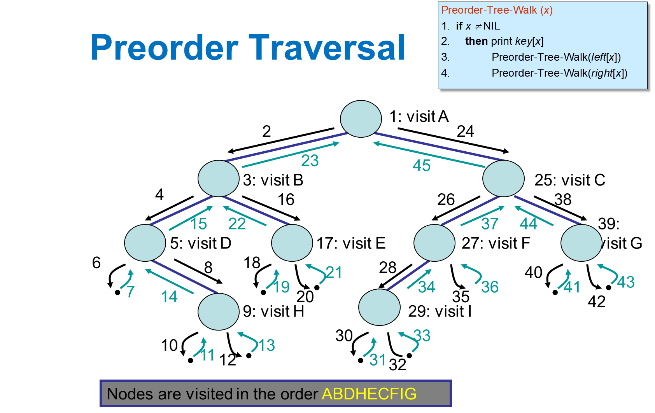
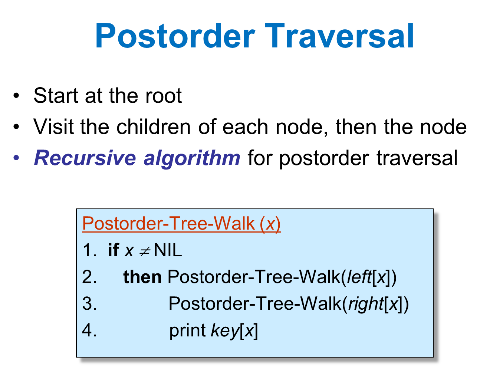
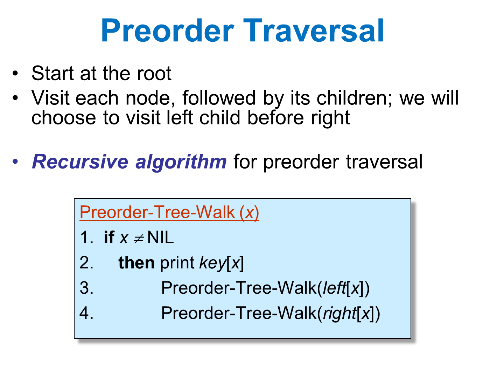
Description automatically generatedText

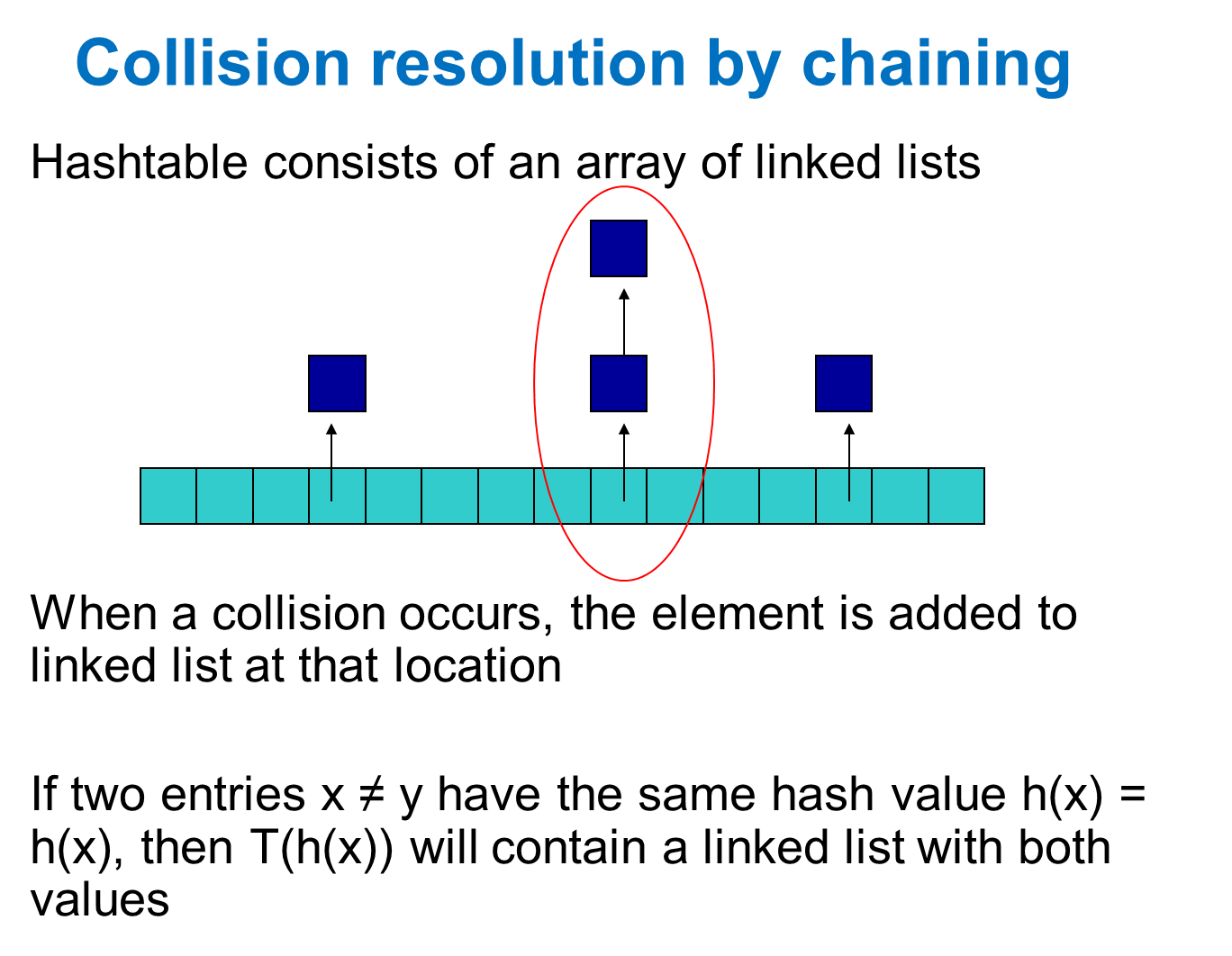
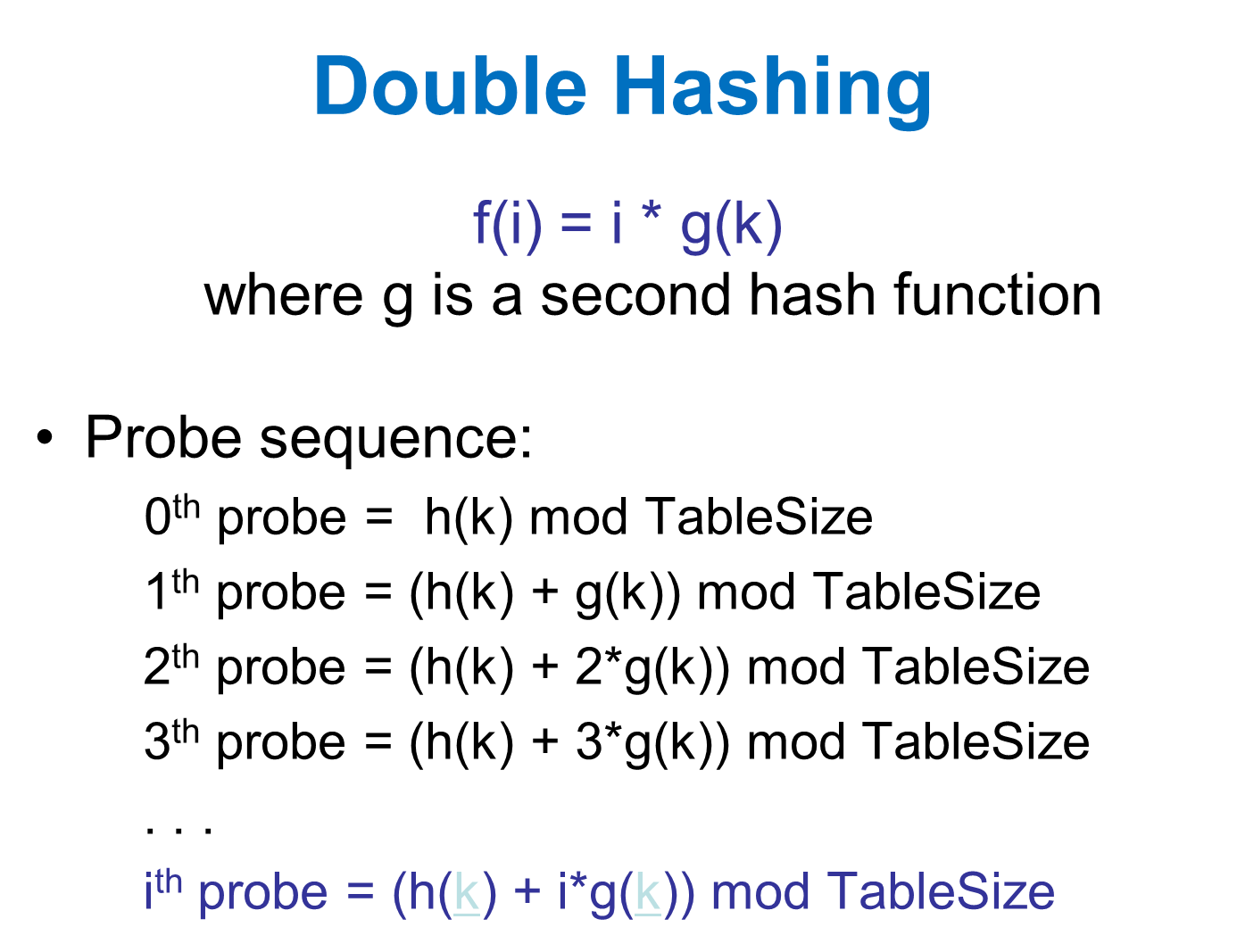
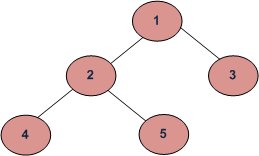
Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generatedShape

Description automatically generated**Graphical user interface, text

Description automatically generated**



Text

Description automatically generated  
(a) Inorder (Left, Root, Right) : 4 2 5 1 3   
(b) Preorder (Root, Left, Right) : 1 2 4 5 3   
(c) Postorder (Left, Right, Root) : 4 5 2 3 1  
Breadth-First or Level Order Traversal: 1 2 3 4 5 