\[ T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2} - 1\right) + c_n \]

\[ T(1) \]

Diagram:

- \( T\left(\frac{n}{2}\right) \)
- \( c_n \)
- \( \frac{n}{2} \)
- \( c(n-1) \)
- \( T(\frac{n}{2}-1) \)
- \( T(\frac{n}{2}) \)
- \( T(n-2) \)

Mathematical expression:

\[ \int_0^n x \, dx = \frac{n^2}{2} \]

Short height:

\[ n \]

Long height:

\[ \log n \]
Tree Delete Example, Page 1

To delete 10 (or any node with 2 children):

1. Find minimum of right subtree
   In this case, the 20
2. Delete this node (delete 20)
3. Transplant 10 with 20
4. Make sure 20's left and right child are set correctly.

To delete 3:

1. Change to NIL
2. Transplant 3 with 1

To delete 7:

1. Replace with the minimum of the right subtree
   In this case, the 9
2. Transplant 7 with 9

Note: The diagram shows a tree with nodes labeled from 1 to 40, 9, and 2. The process of deleting nodes is illustrated with examples of transplanting nodes and changing pointers.
Tree-Delete Example, Page 2

After deleting 4, 3 and 10

Delete 5, replace 5 with 7
Delete 40, replace 40 with 48
**Tree Delete**

**TREE-DELETE**($T$, $z$)

1. if $z\.left == NIL$
2. \hspace{1cm} **TRANSPLANT**($T$, $z$, $z\.right$)
3. \hspace{1cm} elseif $z\.right == NIL$
4. \hspace{1cm} \hspace{1cm} **TRANSPLANT**($T$, $z$, $z\.left$)
5. \hspace{1cm} elseif $y = **TREE-MINIMUM**($z\.right$)$
6. \hspace{1cm} \hspace{1cm} \hspace{1cm} if $y\.p \neq z$
7. \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} **TRANSPLANT**($T$, $y$, $y\.right$)
8. \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} $y\.right = z\.right$
9. \hspace{1cm} \hspace{1cm} \hspace{1cm} \hspace{1cm} $y\.right\.p = y$
10. \hspace{1cm} \hspace{1cm} \hspace{1cm} **TRANSPLANT**($T$, $z$, $y$)

11. \hspace{1cm} $y\.left = z\.left$
12. \hspace{1cm} $y\.left\.p = y$
13. \hspace{1cm} $y\.right = z\.right$
14. \hspace{1cm} $y\.right\.p = y$

**Ω(1) and O(h)**

*Binary Search Trees – 11*