Today: BST, Quick Sort

BST with n nodes has d bonded by \( \log n \leq d \leq n - 1 \)

Question: What’s the cost of building a BST with n elements? Assume the order is equally likely.

Theorem: The expected cost of building a BST with n elements is \( O(n \log n) \)

\( C(n) \): expected number of comparison of building a BST with n elements

\[
c(n) = \sum_{i=1}^{n} \frac{1}{n} ((n - 1) + c(j - 1) + c(n - j))
\]

\[
n * c(n) = \sum_{i=1}^{n} (n - 1) + \sum_{i=1}^{n} c(j - 1) + \sum_{i=1}^{n} c(n - j) = n * (n - 1) + 2 * \sum_{k=0}^{n-1} c(k)
\]

Assume \( c(k) = O(k \log(k)) \) for any \( k < \) so

\[
c(k) = O(k \log k) \leq \alpha * k \log k
\]

\[
\therefore n * c(n) = n * (n - 1) + 2 * \sum_{k=0}^{n-1} \alpha * k \log k \leq n * (n - 1) + 2 * \int_{1}^{n} \alpha * k \log k \, dk
\]

\[
= n * (n - 1) + \alpha * \left( \frac{k^2 \log k - \frac{k^2}{2}}{1} \right) |_{1}^{n}
= n * (n - 1) + \alpha * \left( \frac{n^2 \log n - \frac{n^2}{2} - 0 + \frac{1}{2}}{2} \right)
= \alpha * n^2 \log n - \frac{\alpha}{2} * n^2 - n + \alpha
\]

\[
\therefore c(n) \leq \alpha * n \log n - \frac{\alpha}{2} * n - 1 + \frac{\alpha}{n}
\]

Suppose \( \alpha = 2 \),

\[
c(n) \leq \alpha * n \log n
\]
Base on divide-and-conquer paradigm, sort the array

1. Choose a pivot
   ![Pivot]
   
   **Comment:** quick-sort selects the last element as pivot

2. Divide-and-conquer
   
   **Pivot**
   
   \[ \leq \text{pivot} \quad \geq \text{pivot} \]

Given array A:
1. Divide step: rearrange A into 2 parts such that all elements in the first part are lower than the pivot and that in the last part are larger than the pivot.
2. Conquer step sort the two subarrays recursively using quicksort.

```plaintext
QuickSort(A, p, r)
if p < r
    q < partition(A, p, r)
    QuickSort(A, p, q-1)
    QuickSort(A, q+1, r)
```

```plaintext
Partition(A, p, r)
    x <- A(r)
    i <- p-1
    for j = p to r-1
        if A[j] <= x
            i <- i+1
            swap(A[i], A[j])
    swap(A[i], A[r])
return i+1
```

10 4 8 18 9 1 3 7 6
1. p = 1, r = 9
10 4 8 18 9 1 3 7 6
i j
Loop invariant:
1. All elements with indices $k \leq i$ are lower than pivot.
2. All elements $i < k < j$ are larger than pivot.

Initialization:
- When $j < p$, there is no data to compare

Maintenance:
- If $A[j] > \text{pivot}$, $j \leftarrow j + 1$
- The current array meets the condition 1 & 2
- If $A[j] < \text{pivot}$, we exchange the first element after $i$ with $A[j]$
- The current array meets the condition 1 & 2

Termination:
- When $j = r$
- The array meets the condition 1 & 2

Cost:

Worst case: when the dividing is extremely unbalanced

$T(n) = T(n-1) + cn = \ldots = T(n-i) + \sum_{j=0}^{i-1} c \cdot (n-j) = \ldots = d + \sum_{k=1}^{n} c \cdot (k-1) = d + c \cdot \frac{n \cdot (n+1)}{2} - 1 = \Theta(n^2)$

Best case: when the dividing is perfectly balanced

$T(n) \leq 2 \cdot T\left(\frac{n}{2}\right) + cn = \Theta(n \cdot \log n)(a = 2, b = 2, f(n) = n)$