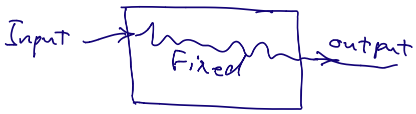
Deterministic Algorithms

- For a specific input, always takes the Same Path to get the output



⇒ for a specific input
↳ fixed Output
fixed Runtime

Randomized Algorithms

(A) Las Vegas Algorithms

* fixed output (always Correct output)
* Randomized Runtime: for a fixed input, takes different routes to generate the output
  ↳ The runtime is different for different Times Runs

(B) Monte Carlo Algorithms

* Fixed Runtime
  ↳ different runs over the Same input take the Same amount of time
* Randomized output: may generate the correct answer. we measure the quality of Alg. by the Probability of generating Correct answer.

A las vegas Randomized Alg. for Quick Sort

Deterministic Q-Sort:

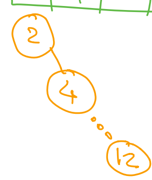
Q-Sort$(X=\{x_1 \cdots x_n\})$

$j=$ Partition (X, x_1) // x_1 is the Pivot
Q-Sort$(\{x_1, \dots, x_{j-1}\})$
Q-Sort $(\{x_{j+1}, \dots, x_n\})$

Issue: The runtime of Q-Sort depends on the initial ordering of elements in the list

e.g.1 (PreSorted)

2	4	5	7	8	9	11	12



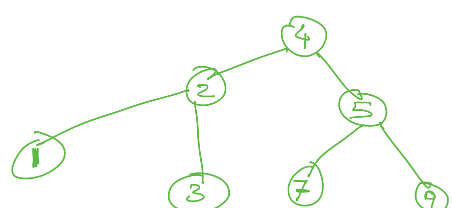
n Comparisons
n-1 ~

$\Rightarrow$ Runtime $= \sum_{i=1}^{n-1} i = O(n^2)$

worste Case

eg. 2:

4	2	5	1	3	7	9



⇒ at each level of the tree the function Partition is Called 2^i times, each of Cost 2^{k-i} $(k=\log n)$. Depth of Tree $= \log n$

$\Rightarrow$ Runtime $= \sum_{i=1}^{k} 2^i 2^{k-i} = n \log n$

Randomized Q-Sort:

Resolves the issue of Q-Sort

Rand Q-Sort($X = \{x_1, \dots, x_n\}$)

- $i = U[1, n]$ // a random Uniform # in range // 1 and n
- $j =$ Partition(X, x_i) // use x_i as Pivot
- Rand Q-Sort($\{x_1 \cdots x_{j-1}\}$)
- Rand Q-Sort($\{x_{j+1} \cdots x_n\}$)

(A) It is clear that the alg. always generates the correct answer (sorted list)

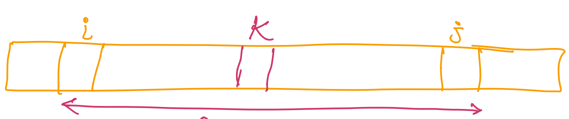
(B) Runtime: Total # of Comparisons derive the Runtime

Let α_{ij} be a random bernouli variable

$$\alpha_{ij} = \begin{cases} 1 & \text{if (in the sorted list) } x_i \text{ and } x_j \text{ have been compared} \\ 0 & \text{otherwise} \end{cases}$$

let P_{ij} be the prob. that $\alpha_{ij} = 1$

Consider the Sort List $x_1, x_2, \dots, x_n$ we want to compute the prob. that in a single run x_i and x_j get compared.



observation: x_i and x_j get compared only if one of them become the pivot, before any x_k between them is selected as Pivot.

If x_k ($i < k < j$) is selected as Pivot first, x_i and x_j fall into separate partitions and never get compared

$$\Rightarrow P_{ij} = \frac{2}{j - i + 1}$$

* for a specific Run, Runtime is:

$$T = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \alpha_{ij}$$

⇒ The expected Runtime is

$$E[T] = E\left[\sum \sum \alpha_{ij}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[\alpha_{ij}]$$

$$E[\alpha_{ij}] = 1 \times P_{ij} + 0 \times (1 - P_{ij}) = P_{ij} = \frac{2}{j - i + 1}$$

$$\Rightarrow E[T] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j - i + 1} = 2 \sum_{i=1}^{n-1} \sum_{j=2}^{i} \frac{1}{i} = 2 \sum_{i=1}^{n-1} H_i$$

$$\boxed{= O(n \log n)}$$