Recap: Probabilities, conditioning and expectations

- Randomized Quicksort
- Quick Selection

- Idea: algorithm that tosses coins and make some decisions based on randomness.
- Why? Simpler algorithm, avoid bad cases

- Random variable

  - Variables that do not have fixed value
  - Example: \( X = \begin{cases} 1 & \text{coin flip is head} \\ 0 & \text{coin flip is tail} \end{cases} \)
  - \( Y = 1, 2, 3, \ldots, 6 \) roll a die
  - \( \Pr[X = i] \): probability that \( X = i \)
    \[ \Pr[X = 0] = \Pr[X = 1] = \frac{1}{2} \]
    \[ \Pr[Y = 1] = \Pr[Y = 2] = \ldots = \Pr[Y = 6] = \frac{1}{6} \]
  - Example: \( Z \): sum of points for 2 dice
    \[ \Pr[Z = 2] = \frac{1}{36}, \quad \Pr[Z = 7] = \frac{1}{6} \]

- Expectation

  - Similar to "average"
    \[ \mathbb{E}[X] = \sum_i \Pr[X = i] \cdot i \]
    \[ \mathbb{E}[X] = \frac{1}{2}, \quad \mathbb{E}[Y] = 3.5 \]
  - Linearity of expectation
    \[ \mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y] \]
    \[ Y \text{ die, } Y' \text{ another die, } Z = Y + Y' \]
    \[ \mathbb{E}[Z] = \sum_{i=2}^{12} \Pr[Z = i] \cdot i \]
    \[ \mathbb{E}[Z] = \mathbb{E}[Y] + \mathbb{E}[Y'] = 7 \]

- Conditioning

  - \( \Pr[Z = i \mid Y = j] \): probability that \( Z = i \), given we already know \( Y = j \)
  - \( \Pr[Z = 6 \mid Y = 3] \): We know first die is 3, what is the probability
that sum of two dice is 6.

\[ \Pr[Z = 6 | Y = 3] = \frac{1}{6} \quad \Pr[Z = 6] = \frac{5}{36} \]

\[ \Pr[Z = 2 | Y = 3] = 0 \quad \Pr[Z = 2] = \frac{1}{36} \]

\[ \Pr[Y' = 3 | Y = 4] = \Pr[Y' = 3] = \frac{1}{6} \]

\[ Y', Y' \text{ are independent.} \]

- Joint probability: two events happen simultaneously

\[ \Pr[Y = 3, Y' = 4] : \text{probability that first die is 3, second is 4} \]

\[ \Pr[Y = 3, Y' = 4] = \frac{1}{36} = \Pr[Y = 3] \cdot \Pr[Y' = 4 | Y = 3] \]

\[ \Pr[X = i, Y = j] = \Pr[X = i] \cdot \Pr[Y = j | X = i] \]

\[ \Pr[Y = j | X = i] = \frac{\Pr[X = i, Y = j]}{\Pr[X = i]} \]

\[ \Pr[Y = 3, Z = 4] = \frac{1}{36} = \Pr[Y = 3] \cdot \Pr[Z = 4 | Y = 3] \]

\[ \neq \Pr[Y = 3] \cdot \Pr[Z = 4] \]

\[ \Pr[X_n = a_n, X_{n-1} = a_{n-1}, \ldots, X_i = a_i] = \Pr[X_n = a_n] \cdot \Pr[X_{n-1} = a_{n-1} | X_n = a_n] \]

\[ \cdot \Pr[X_{n-2} = a_{n-2} | X_n = a_n, X_{n-1} = a_{n-1}] \]

\[ \cdot \Pr[X_k = a_k | X_n = a_n, X_{n-1} = a_{n-1}, \ldots, X_i = a_i] \]

- \[ E[Z | Y = i] = \sum_j \Pr[Z = j | Y = i] \cdot j \quad \text{(conditioned expectation)} \]

\[ E[Z | Y = 2] = 5.5 = E[Y | Y = 2] + E[Y | Y = 2] \]

\[ = 2 + 3.5 \]

- Law of total expectation

\[ E[Y] = \sum \Pr[X = i] \cdot E[Y | X = i] \]

\[ \text{running time of a randomized algorithm} \quad \text{first random choice made by the algorithm} \quad \text{expected running time after we fix first random choice} \]

- Quicksort

- Pick a pivot

\[ 5 \leq 4 \ 8 \ 7 \ 6 \ 9 \ 1 \]

- Partition the array into smaller and larger parts

\[ 5 \leq 4 \ 1 \ 6 \ 8 \ 7 \ 9 \]
- Recurse on left and right parts.
- Running time \( O(n \log n) \times x \): in worst case running time is \( \Theta(n^2) \)
- Problem: if every time pivot is the smallest/largest, then quicksort takes \( \Theta(n^3) \) time.
- Idea: pick a random pivot number.
- How to analyze?

\( T_n \): random variable: running time of randomized quicksort for array of size \( n \).

Want: \( \mathbb{E}[T_n] \) is small.

\[
T_n = \begin{cases} 
T_{n-1} + n & \text{if pivot is smallest} \quad \Pr = \frac{1}{n} \\
T_1 + T_{n-2} + n & \text{pivot is second smallest} \quad \Pr = \frac{1}{n} \\
T_{i-1} + T_{n-1} + n & \text{pivot is } i^{th} \text{ smallest} \quad \Pr = \frac{1}{n}
\end{cases}
\]

\[
\mathbb{E}[T_n] = \sum_{i=1}^{n} \frac{1}{n} \left( \mathbb{E}[T_{i-1} + T_{n-1} + n] \right)
= n + \sum_{i=1}^{n} \left( \mathbb{E}[T_{i-1}] + \mathbb{E}[T_{n-1}] \right)
\]

Next time: will show \( \mathbb{E}[T_n] \leq 4n \log_2 n \)