Dynamic Programming Part 2: Probability, Combinatorics, and Bitmasks

Duke Compsci 309s

Siyang Chen

Introduction

Probability, combinatorics, and bitmasking appear commonly in dynamic programming problems.

Kolmogorov's axioms of probability

- The probability $P(A)$ of an event A is a nonnegative real number.
- The sum of the probabilities of all atomic events is 1.
- If A and B are mutually exclusive, then $P(A \cup B) = P(A) + P(B)$.

Kolmogorov's axioms of probability

- The probability $P(A)$ of an event A is a nonnegative real number.
- The sum of the probabilities of all atomic events is 1.
- If A and B are mutually exclusive, then $P(A \cup B) = P(A) + P(B)$.

I'm assuming everyone has a basic understanding of probability, so we won't dwell on these here.

Useful properties

- Independence: Two events A and B are called *independent* if

$$P(A)P(B) = P(A \cap B)$$

Useful properties

- Independence: Two events A and B are called *independent* if

$$P(A)P(B) = P(A \cap B)$$

- Linearity of expectation: If a random variable A can take on some values $a_1, a_2, \ldots, a_n$, then its *expected value* is

$$\mathbb{E}(A) = \sum_{i=1}^{n} a_i P(A = a_i)$$

Useful properties

- Independence: Two events A and B are called *independent* if

$$P(A)P(B) = P(A \cap B)$$

- Linearity of expectation: If a random variable A can take on some values $a_1, a_2, \ldots, a_n$, then its *expected value* is

$$\mathbb{E}(A) = \sum_{i=1}^{n} a_i P(A = a_i)$$

  Expectation is *linear*, i.e.:

$$\mathbb{E}(A + cB) = \mathbb{E}(A) + c\mathbb{E}(B)$$

  Note that this is true even for events that aren't independent!

Example problem: CoinReversing

Look at the problem `CoinReversing` on the syllabus.

Example problem: CoinReversing

Represent a coin's end state by

$$C_i = \begin{cases} 1 & \text{if heads} \\ 0 & \text{if tails} \end{cases}$$

Example problem: CoinReversing

Represent a coin's end state by

$$C_i = \begin{cases} 1 & \text{if heads} \\ 0 & \text{if tails} \end{cases}$$

We want to compute

$$\mathbb{E}(C_1 + C_2 + \ldots + C_N)$$

Example problem: CoinReversing

Represent a coin's end state by

$$C_i = \begin{cases} 1 & \text{if heads} \\ 0 & \text{if tails} \end{cases}$$

We want to compute

$$\mathbb{E}(C_1 + C_2 + \ldots + C_N)$$

By linearity of expectation, this is

$$\mathbb{E}(C_1) + \mathbb{E}(C_2) + \ldots + \mathbb{E}(C_N)$$

Example problem: CoinReversing

Represent a coin's end state by

$$C_i = \begin{cases} 1 & \text{if heads} \\ 0 & \text{if tails} \end{cases}$$

We want to compute

$$\mathbb{E}(C_1 + C_2 + \ldots + C_N)$$

By linearity of expectation, this is

$$\mathbb{E}(C_1) + \mathbb{E}(C_2) + \ldots + \mathbb{E}(C_N)$$

Since the coins are the same, this is

$$N \times \mathbb{E}(C)$$

Example problem: CoinReversing

- Now we want to compute $\mathbb{E}(C)$, where C is 1 if the coin is heads after the K flips and 0 otherwise.

Example problem: CoinReversing

- Now we want to compute $\mathbb{E}(C)$, where C is 1 if the coin is heads after the K flips and 0 otherwise.
- Let p_i be the probability that the coin is heads up after i flips. We are given that $p_0 = 1$; $\mathbb{E}(C) = p_K$.

Example problem: CoinReversing

- Now we want to compute $\mathbb{E}(C)$, where C is 1 if the coin is heads after the K flips and 0 otherwise.
- Let p_i be the probability that the coin is heads up after i flips. We are given that $p_0 = 1$; $\mathbb{E}(C) = p_K$.
- Let f_i be the probability that some coin is flipped on the i-th step:

Example problem: CoinReversing

- Now we want to compute $\mathbb{E}(C)$, where C is 1 if the coin is heads after the K flips and 0 otherwise.
- Let p_i be the probability that the coin is heads up after i flips. We are given that $p_0 = 1$; $\mathbb{E}(C) = p_K$.
- Let f_i be the probability that some coin is flipped on the i-th step: $f_i = a_i/N$.

Example problem: CoinReversing

- Now we want to compute $\mathbb{E}(C)$, where C is 1 if the coin is heads after the K flips and 0 otherwise.
- Let p_i be the probability that the coin is heads up after i flips. We are given that $p_0 = 1$; $\mathbb{E}(C) = p_K$.
- Let f_i be the probability that some coin is flipped on the i-th step: $f_i = a_i/N$.
- Since the probability a coin gets flipped at each point in time is independent, then

$$p_{i+1} = p_i \times (1 - f_i) + (1 - p_i) \times f_i$$

Example problem: CoinReversing

- Now we want to compute $\mathbb{E}(C)$, where C is 1 if the coin is heads after the K flips and 0 otherwise.
- Let p_i be the probability that the coin is heads up after i flips. We are given that $p_0 = 1$; $\mathbb{E}(C) = p_K$.
- Let f_i be the probability that some coin is flipped on the i-th step: $f_i = a_i/N$.
- Since the probability a coin gets flipped at each point in time is independent, then

$$p_{i+1} = p_i \times (1 - f_i) + (1 - p_i) \times f_i$$

Java solution:
`https://github.com/md143rbh7f/competitions/blob/master/topcoder/srm/518/CoinReversing.java`

Bitmasking

Bitmasking is a compact and efficient way to represent sets as the bits in an integer.

Bitmasking

Bitmasking is a compact and efficient way to represent sets as the bits in an integer.
The i-th bit of S is on if and only if element i is included in S.

Bitmasking

Bitmasking is a compact and efficient way to represent sets as the bits in an integer.
The i-th bit of S is on if and only if element i is included in S.
E.g.:

Bitmasking

Bitmasking is a compact and efficient way to represent sets as the bits in an integer.
The i-th bit of S is on if and only if element i is included in S.
E.g.:

- $8 = 111_2 = \{0, 1, 2\}$

Bitmasking

Bitmasking is a compact and efficient way to represent sets as the bits in an integer.
The i-th bit of S is on if and only if element i is included in S.
E.g.:

- $8 = 111_2 = \{0, 1, 2\}$
- $41 = 101001_2 = \{0, 3, 5\}$

Bitmasking

Bitmasking is a compact and efficient way to represent sets as the bits in an integer.
The i-th bit of S is on if and only if element i is included in S.
E.g.:

- $8 = 111_2 = \{0, 1, 2\}$
- $41 = 101001_2 = \{0, 3, 5\}$
- $28 = 11100_2 = \{2, 3, 4\}$

Bitmasking

Bitmasking is a compact and efficient way to represent sets as the bits in an integer.
The i-th bit of S is on if and only if element i is included in S.
E.g.:

- $8 = 111_2 = \{0, 1, 2\}$
- $41 = 101001_2 = \{0, 3, 5\}$
- $28 = 11100_2 = \{2, 3, 4\}$
- Etc...

Bitmasking

To iterate over all sets of size N, simply iterate from 0 to $2^N - 1$:

$$S = 000$$
$$S = 001$$
$$S = 010$$
$$S = 011$$
$$\dots$$
$$S = 110$$
$$S = 111$$

Bit operations

- To shift x left by i bits: `x << i`

Bit operations

- To shift x left by i bits: `x << i`
- … or right: `x >> i`

Bit operations

- To shift x left by i bits: `x << i`
- ... or right: `x >> i`
- To check if the i-th bit is on: `(x & (1 << i)) > 0`

Bit operations

- To shift x left by i bits: `x << i`
- ... or right: `x >> i`
- To check if the i-th bit is on: `(x & (1 << i)) > 0`
- To flip the i-th bit: `x ^ (1 << i)`

Bit operations

- To shift x left by i bits: `x << i`
- ... or right: `x >> i`
- To check if the i-th bit is on: `(x & (1 << i)) > 0`
- To flip the i-th bit: `x ^ (1 << i)`
- To turn the i-th bit on: `x | (1 << i)`

Example problem: Fish

Look at the problem `Fish` on the syllabus.

Combinatorics