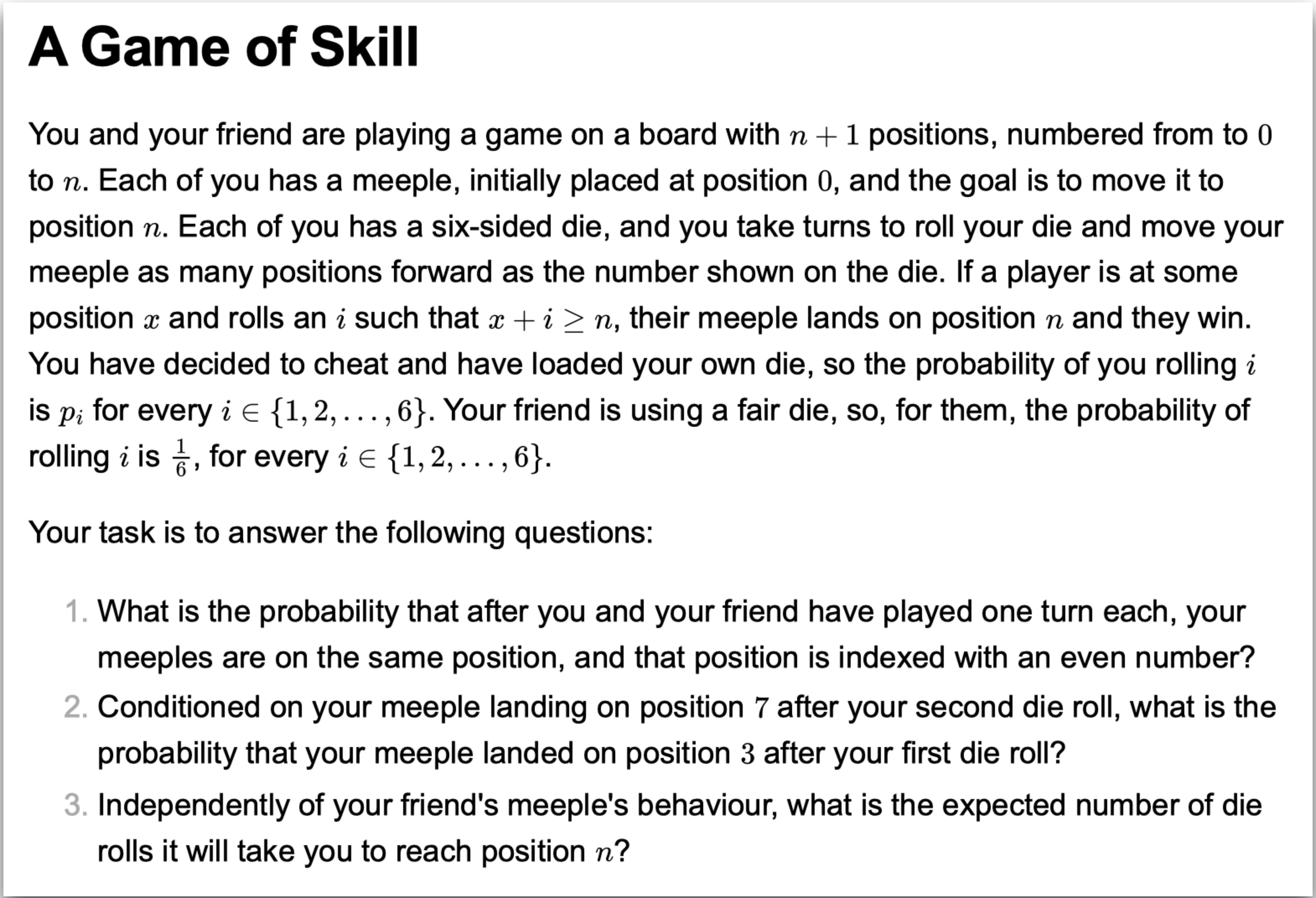
Algorithms and Probability

Week 14

Georg Hasebe — 30/05/2024



A Game of Skill

You and your friend are playing a game on a board with $n+1$ positions, numbered from to 0 to n. Each of you has a meeple, initially placed at position 0, and the goal is to move it to position n. Each of you has a six-sided die, and you take turns to roll your die and move your meeple as many positions forward as the number shown on the die. If a player is at some position x and rolls an i such that $x + i \geq n$, their meeple lands on position n and they win. You have decided to cheat and have loaded your own die, so the probability of you rolling i is p_i for every $i \in \{1, 2, \ldots, 6\}$. Your friend is using a fair die, so, for them, the probability of rolling i is $\frac{1}{6}$, for every $i \in \{1, 2, \ldots, 6\}$.

Your task is to answer the following questions:

1. What is the probability that after you and your friend have played one turn each, your meeples are on the same position, and that position is indexed with an even number?
2. Conditioned on your meeple landing on position 7 after your second die roll, what is the probability that your meeple landed on position 3 after your first die roll?
3. Independently of your friend's meeple's behaviour, what is the expected number of die rolls it will take you to reach position n?

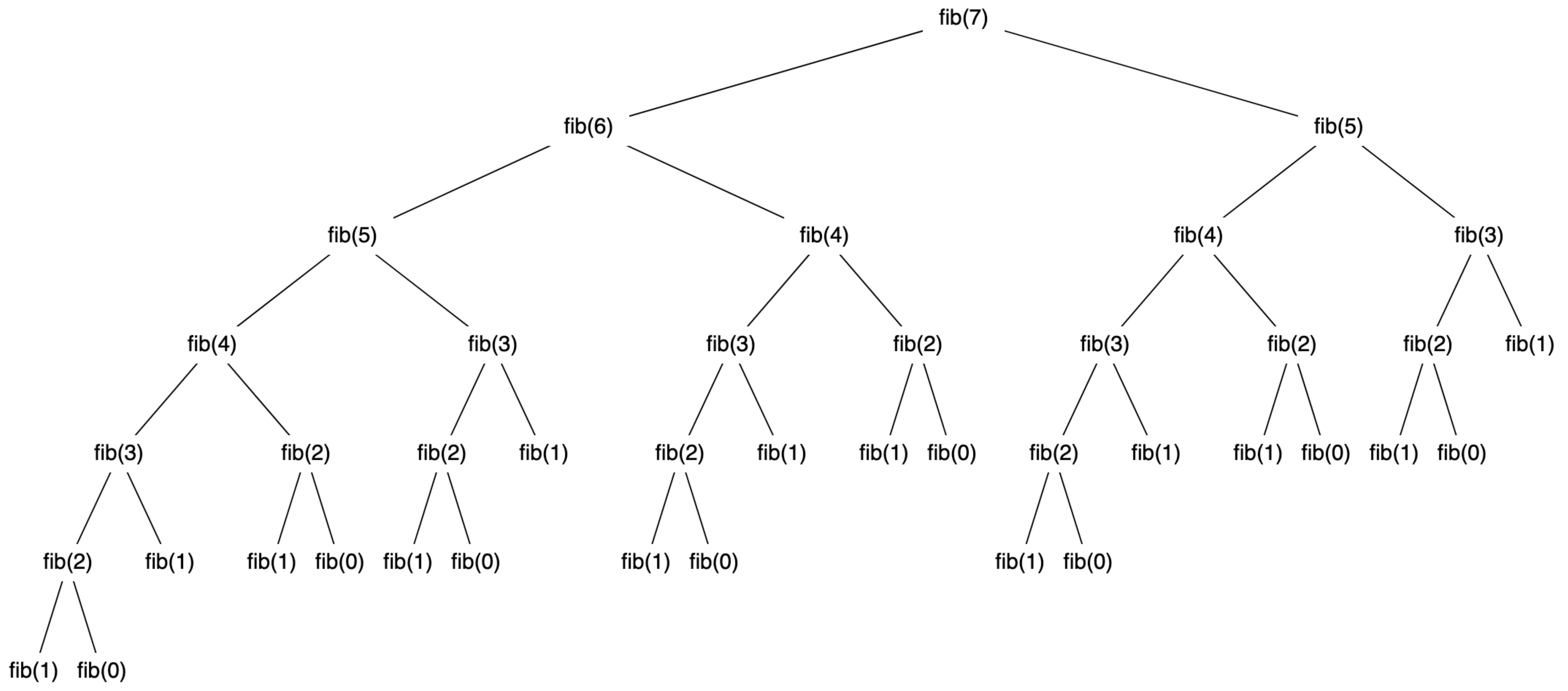
cont'd on iPad.

Remarks

- Read the description **carefully** (multiple times if needed!)
- **Take notes** if needed (on assumptions, values etc.)
- Study **input/output format** and **constraints**
- Study the **sample cases** and try to make sense of them
- **Don’t jump around** between multiple tasks (in my opinion)
- Be aware of how much **time** you have left

Dynamic Programming

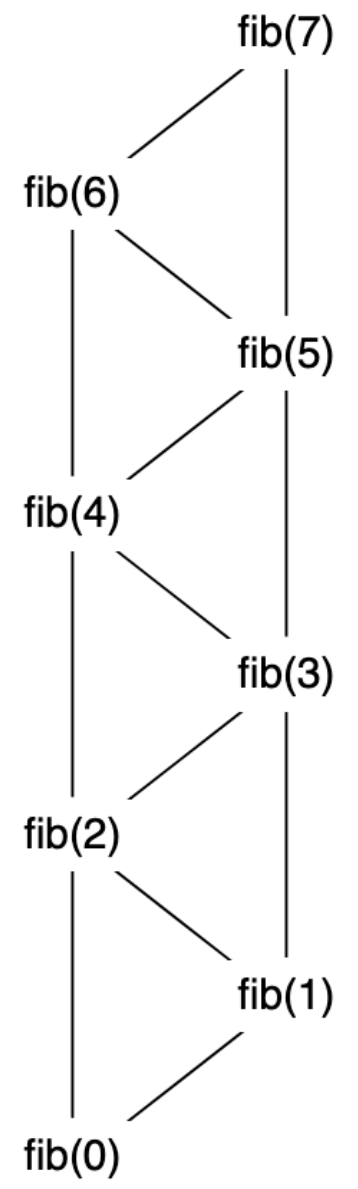
"Dynamic Programming refers to simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner."[1]



Tabulation,
Memoization.

$\longrightarrow$

By reusing results that have already been computed we can reduce the complexity from $O(2^n)$ to $O(n)$.



[1]: https://en.wikipedia.org/wiki/Dynamic_programming
[Image]: Link

Remarks

- Always **think** before starting to code
- **Test** your ideas before starting to code (e.g. on sample cases)
- Don’t get stuck on one approach
- I highly recommend watching the DP video on Moodle.

Recurrence

We define

$$f(x) := \begin{cases} \mathbb{E}[\text{"\# of rolls from } x \text{ to } n\text{"}] & 0 \leq x \leq n, \\ 0 & x > n\,. \end{cases}.$$

We derived the following recurrence for $0 \leq x < n$

$$f(x) = \sum_{i=1}^{6} p_i \cdot (1 + f(x+i)) = 1 + \sum_{i=1}^{6} p_i \cdot f(x+i).$$