

CS 4800: Algorithms & Data

Lecture 6

January 27, 2017

Randomized algorithms

Events and probabilities

Suppose you're on a game show, and you're given the choice of three doors (A, B, C). Behind one door is a car, behind the others, goats. You pick a door, say A, and the host, who knows what's behind the doors, opens another door, say C, which has a goat. He says to you, "Do you want to pick door B?" Is it to your advantage to switch your choice of doors?

Assumptions

- Car is equally likely to be behind each door
- Player is equally likely to pick each door
- After player picks, host opens a different door with a goat behind
- If the host has choices, he is equally likely to pick each of them

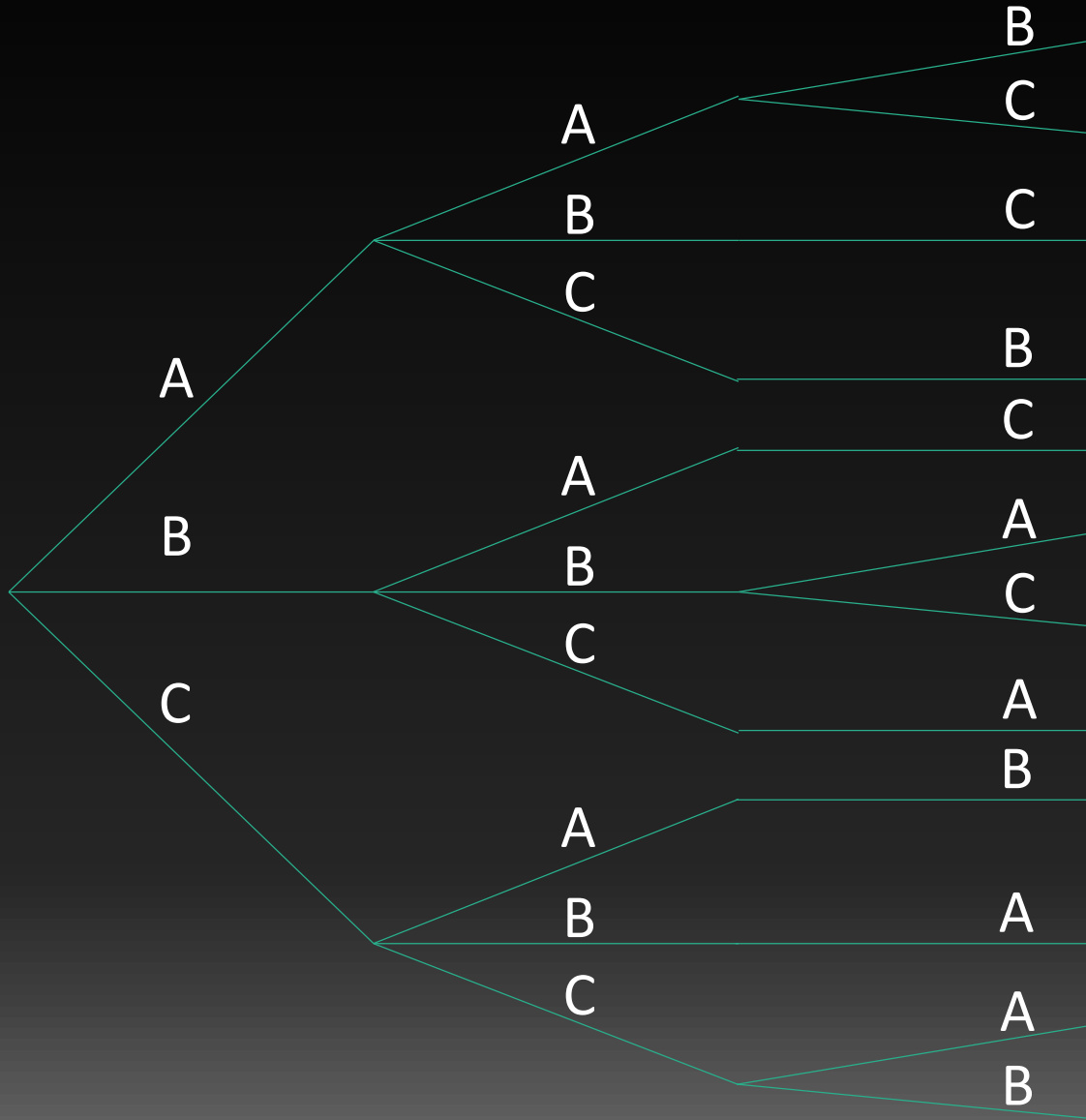
Sample space

- Randomly determined quantities:
 - Car location
 - Door chosen by player
 - Door opened by host
- Every possible combination is an **outcome**
- Set of all outcomes is **sample space**

Car
location

Player's
1st guess

Door
revealed

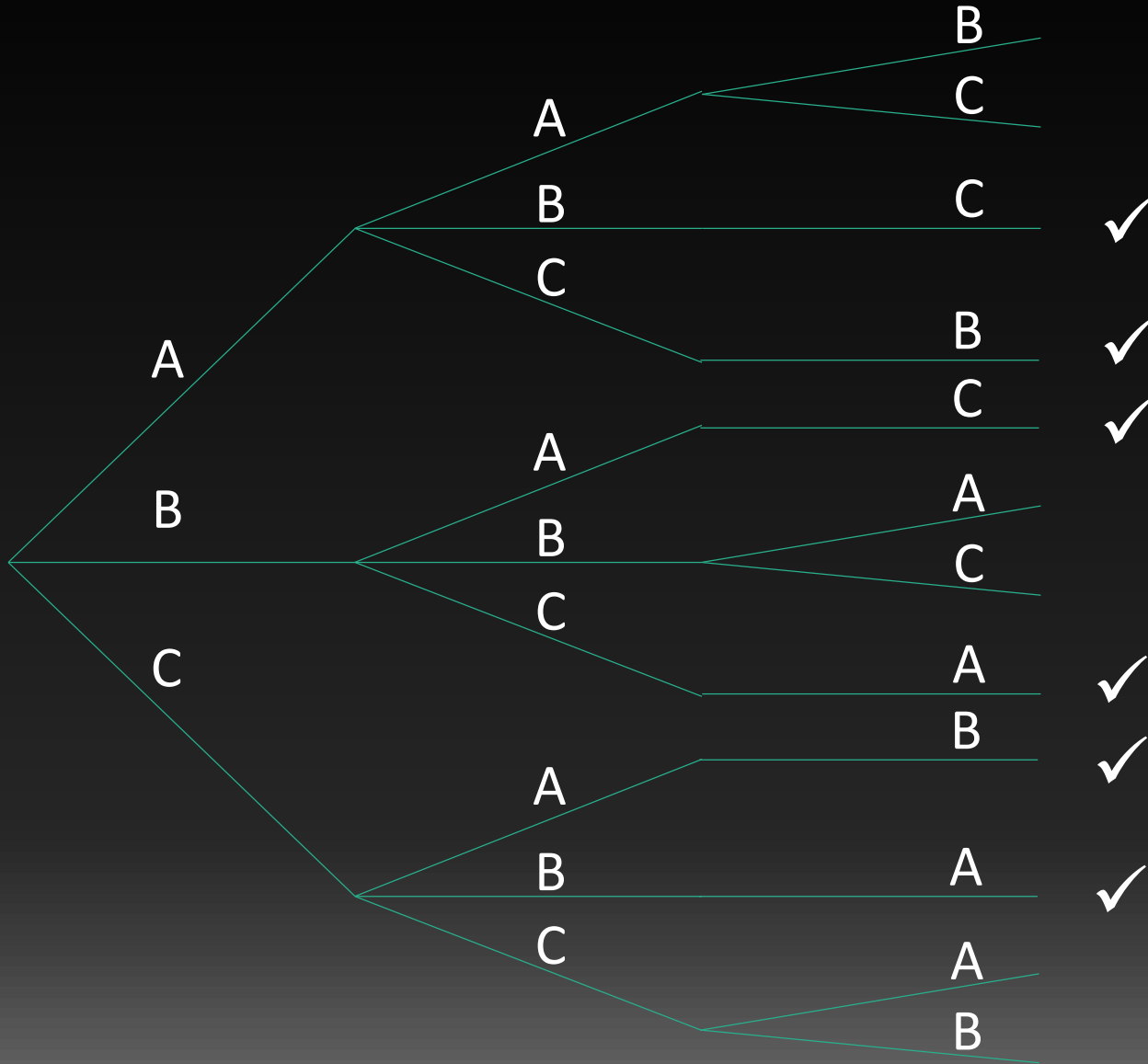


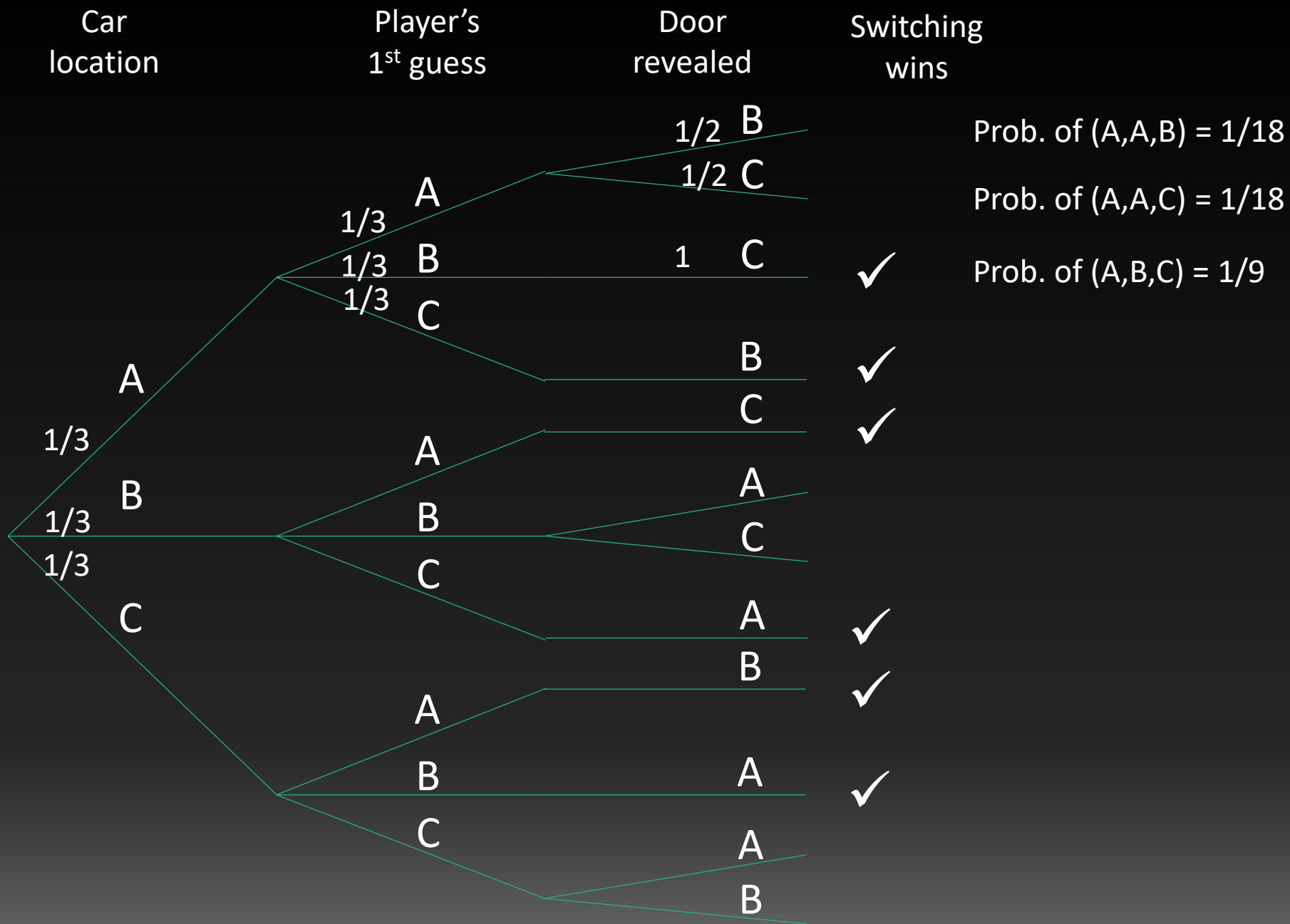
Car location

Player's 1st guess

Door revealed

Switching wins





Random variables

- Random variable R is a function
 $R: \{\text{sample space}\} \rightarrow \mathbb{R}$
- Outcomes of 2 fair coin tosses
- $R = \# \text{heads in the outcome}$
- $R(\text{HH}) = 2$
- $R(\text{HT}) = 1$
- $\Pr[R = 1] = \Pr[\text{HT}] + \Pr[\text{TH}] = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$

Expectation

- R is random variable on sample space S
- $E[R] = \sum_{outcome\ w} R(w)Pr[w]$
- R : #heads in 2 fair coin tosses
- $E[R] = R(TT) Pr[TT] + R(TH) Pr[TH] + R(HT) Pr[HT] + R(HH) Pr[HH]$
- $E[R] = 0 + \frac{1}{4} + \frac{1}{4} + \frac{2}{4} = 1$

Linearity of expectation

Claim. For any variables R_1, R_2 ,

$$E[R_1 + R_2] = E[R_1] + E[R_2]$$

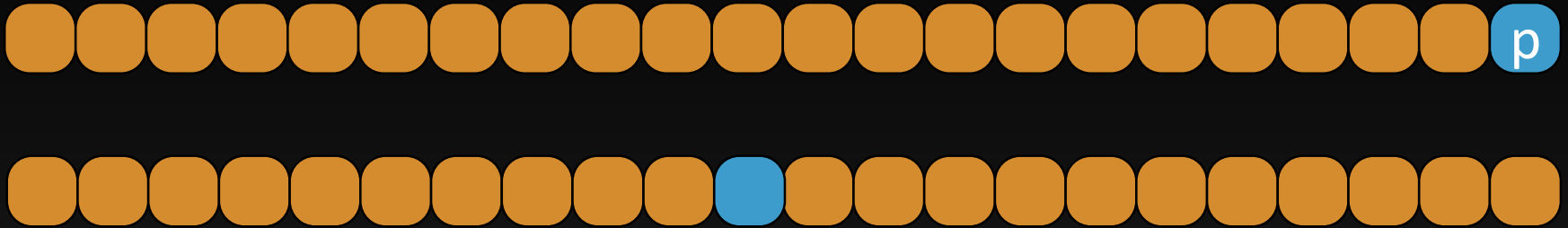
Proof. Let $R = R_1 + R_2$.

$$\begin{aligned} E[R] &= \sum_{w \in \mathcal{S}} R(w) \Pr[w] \\ &= \sum_{w \in \mathcal{S}} (R_1(w) + R_2(w)) \Pr[w] \\ &= \sum_{w \in \mathcal{S}} R_1(w) \Pr[w] + \sum_{w \in \mathcal{S}} R_2(w) \Pr[w] \\ &= E[R_1] + E[R_2] \end{aligned}$$

Application

- R : #heads in 2 fair coin tosses
- $R = R_1 + R_2$ where R_1 =#heads in 1st coin toss
- $E[R_1]=1/2$ (head with probability $\frac{1}{2}$, tail with probability $\frac{1}{2}$)
- $E[R_2]=1/2$
- $E[R] = \frac{1}{2} + \frac{1}{2} = 1$
- $E[\text{\#heads in 100 coin tosses}] = ?$
- 100 coin tosses. $E[\text{\#times where two consecutive coins are different}] = ?$

Quicksort



- Pick an element p
- Partition the list using p as pivot
 - Left half are elements $< p$
 - Right half are elements $> p$
- Recursively sort both halves

How to pick good pivot p ?

Picking good pivot

- Can run median algorithm to use median as pivot
- $O(n)$ time to find pivot
- $T(n) = 2T(n/2) + O(n)$
- Solution?
- Cons: Constant in $O(n)$ is large
- New idea: use random pivot

Running time with random pivot

- Suffices to count number of comparisons
- V_i : i^{th} smallest value in array $A[1\dots n]$
- X_{ij} : random variable that is 1 if we compare V_i and V_j and 0 otherwise
- $\#comparisons = \sum_{i=1}^n \sum_{j=i+1}^n X_{ij}$
- $E[\#comparisons] = \sum_{i=1}^n \sum_{j=i+1}^n E[X_{ij}]$

When do we compare V_i and V_j ?

- If V_k is picked as pivot and $V_i < V_k < V_j$
 - V_i goes left, V_j goes right
 - We do not compare V_i and V_j
- In general, we compare V_i and V_j if and only if the first pivot chosen from $\{V_i, V_{i+1}, \dots, V_j\}$ is either V_i or V_j .
- By symmetry, the probability of this is $\frac{2}{j-i+1}$
- $E[X_{ij}] = \frac{2}{j-i+1}$

Running time of Quicksort

- $E[\#comparisons] = \sum_{i=1}^n \sum_{j=i+1}^n E[X_{ij}]$
$$= \sum_{i=1}^n \sum_{j=i+1}^n \frac{2}{j-i+1}$$
$$= \sum_{k=1}^n \sum_{i=1}^{n-k} \frac{2}{k+1} \quad (\text{reorder sums, } j = i + k)$$
$$= \sum_{k=1}^n \frac{2(n-k)}{k+1}$$
$$= \sum_{k=1}^n \left(\frac{2n+2}{k+1} - 2 \right)$$
$$= (2n+2) \sum_{k=1}^n \frac{1}{k+1} - 2n$$

Harmonic number $< \ln(n+1)$