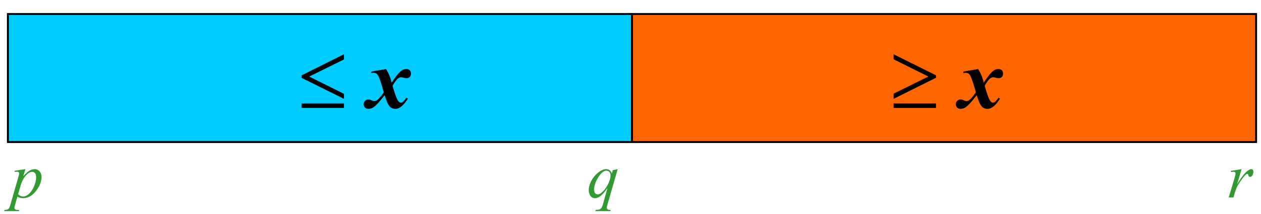
CS473-Algorithms I

Lecture 6-a

Randomized Quicksort

Analysis of Quicksort

QUICKSORT (A, p, r)
    if $p < r$ then
        $q \leftarrow$ H-PARTITION(A, p, r)
        QUICKSORT(A, p, q)
        QUICKSORT(A, q +1, r)



- Assume all elements are distinct
- Let $T(n)$=worst-case running time

Worst-case of quicksort

- Input sorted or reverse sorted.
- Partition around min or max element.
- One side of partition always has one element.

$$
\begin{aligned}
T(n) &= T(1) + T(n-1) + \Theta(n) \\
&= \Theta(1) + T(n-1) + \Theta(n) \\
&= T(n-1) + \Theta(n) \\
&= \Theta(n^2) \qquad \textit{(arithmetic series)}
\end{aligned}
$$

Worst-case recursion tree

$$T(n) = T(1) + T(n–1) + cn$$

Worst-case recursion tree

$$T(n) = T(1) + T(n-1) + cn$$

$$T(n)$$

Worst-case recursion tree

$$T(n) = T(1) + T(n-1) + cn$$

$$cn$$

$$T(1) \qquad T(n-1)$$

Worst-case recursion tree

$$T(n) = T(1) + T(n-1) + cn$$

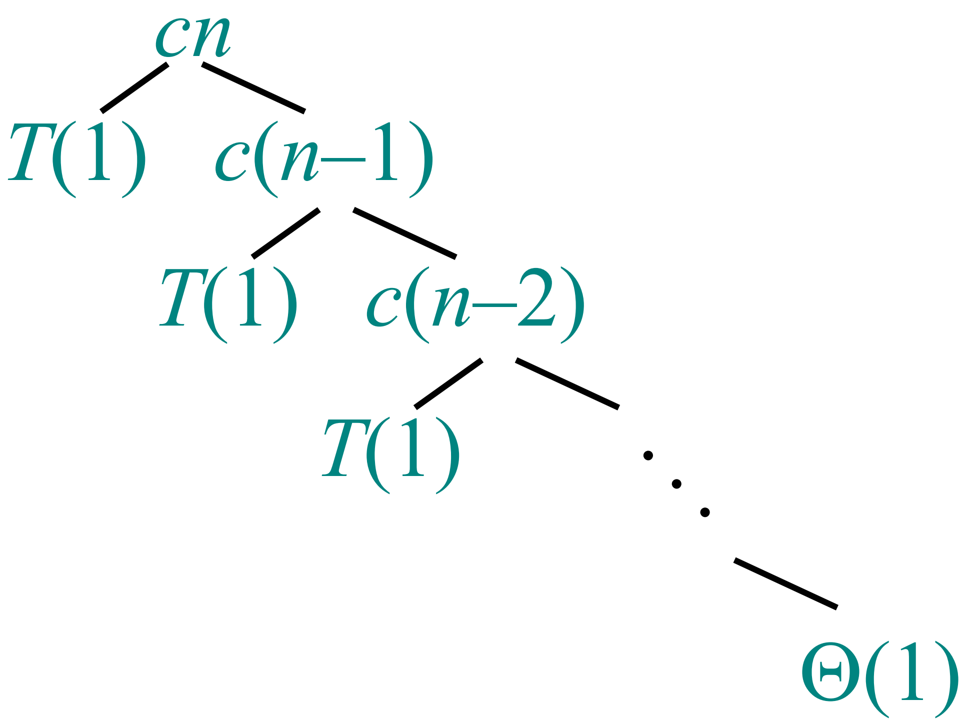
$$cn$$

$$T(1) \qquad c(n-1)$$

$$T(1) \qquad T(n-2)$$

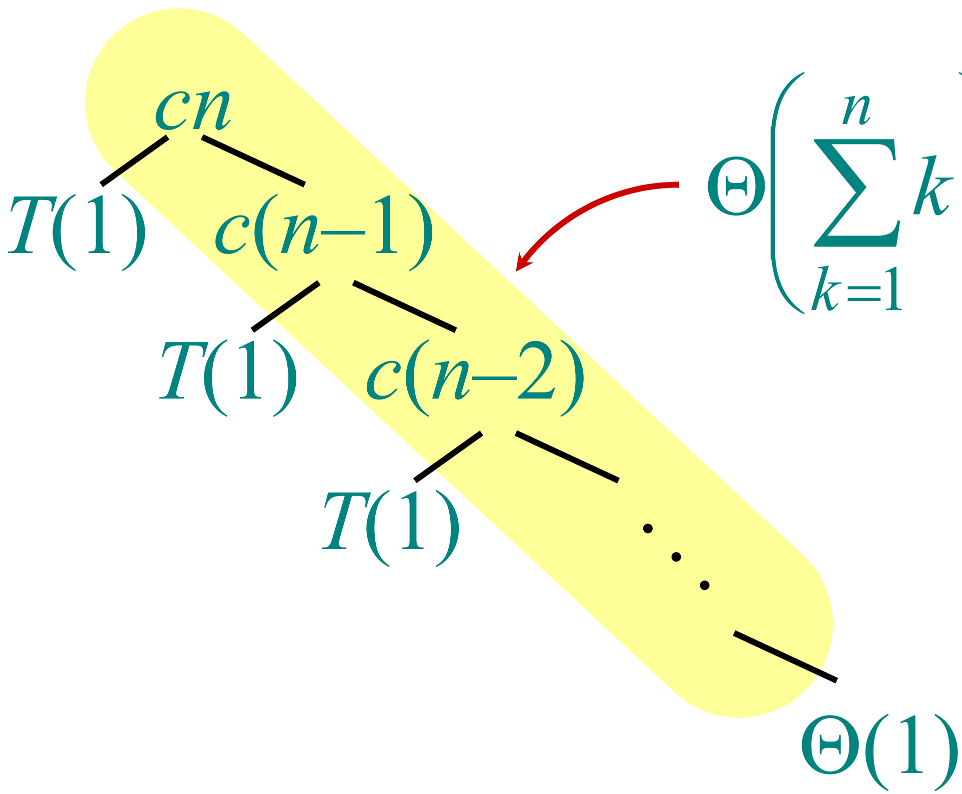
Worst-case recursion tree

$$T(n) = T(1) + T(n-1) + cn$$



Worst-case recursion tree

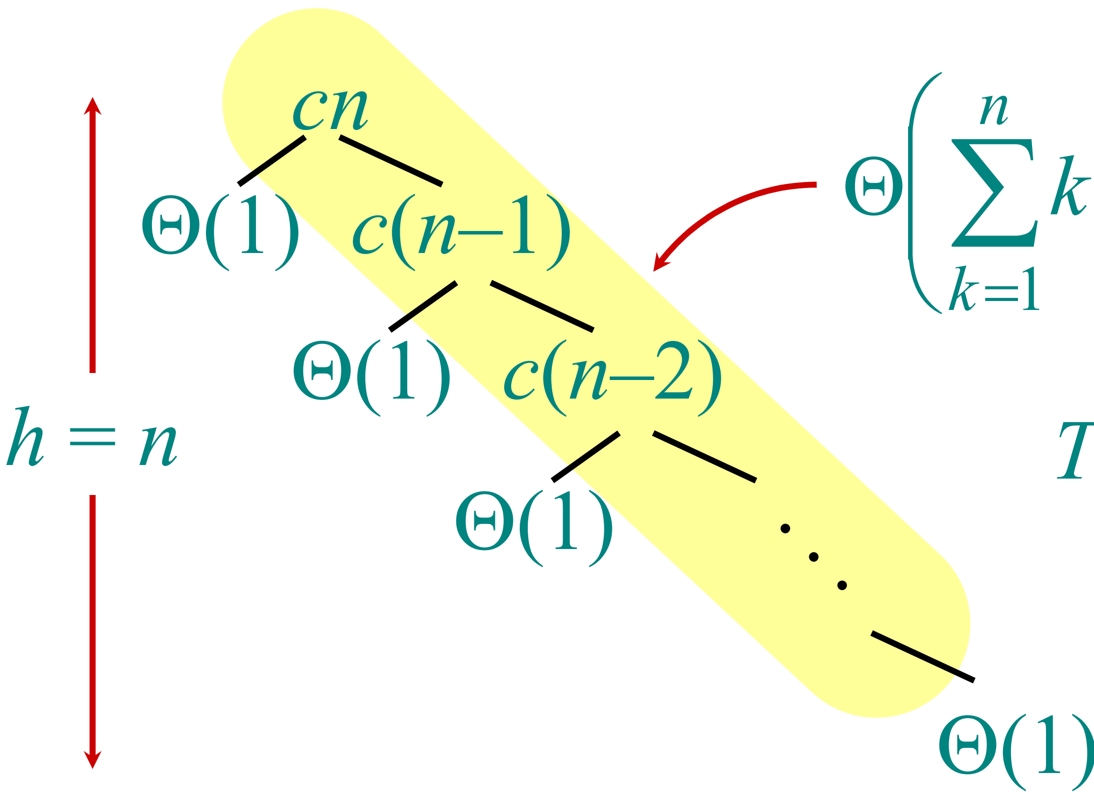
$$T(n) = T(1) + T(n-1) + cn$$



$$\Theta\left(\sum_{k=1}^{n} k\right) = \Theta\left(n^2\right)$$

Worst-case recursion tree

$$T(n) = T(1) + T(n-1) + cn$$



$$\Theta\left(\sum_{k=1}^{n} k\right) = \Theta\left(n^2\right)$$

$$\begin{aligned} T(n) &= \Theta(n) + \Theta(n^2) \\ &= \Theta(n^2) \end{aligned}$$

Best-case analysis

(For intuition only!)

If we're lucky, H-PARTITION splits the array evenly:

$$\begin{aligned} T(n) &= 2T(n/2) + \Theta(n) \\ &= \Theta(n \lg n) \qquad \text{(same as merge sort)} \end{aligned}$$

What if the split is always $\frac{1}{10} : \frac{9}{10}$?

$$T(n) = T\left(\tfrac{1}{10}n\right) + T\left(\tfrac{9}{10}n\right) + \Theta(n)$$

What is the solution to this recurrence?

Analysis of “almost-best” case

$T(n)$

Analysis of “almost-best” case

$$cn$$

$$T\left(\tfrac{1}{10}n\right) \qquad\qquad T\left(\tfrac{9}{10}n\right)$$

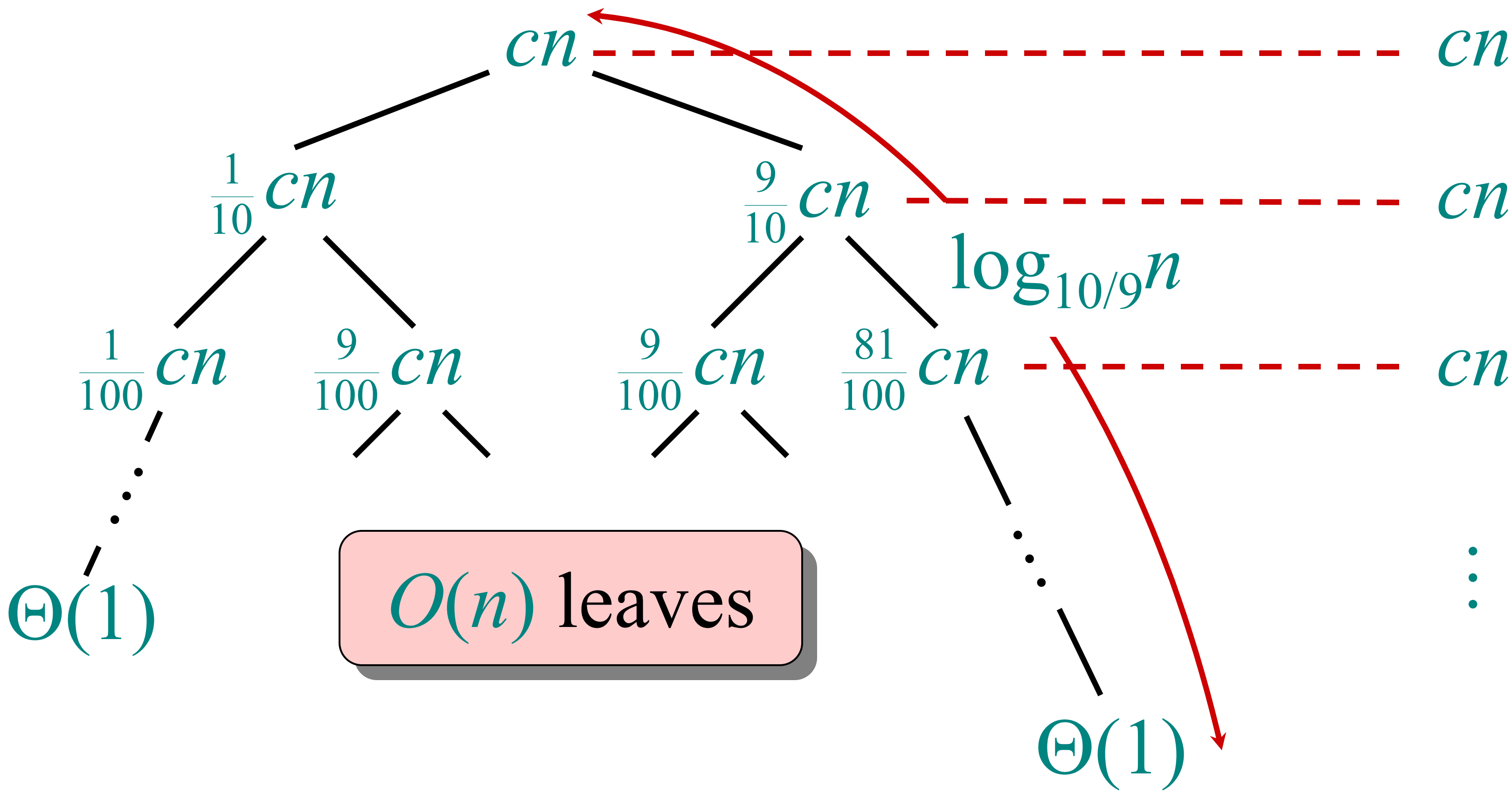
Analysis of “almost-best” case

$$cn$$

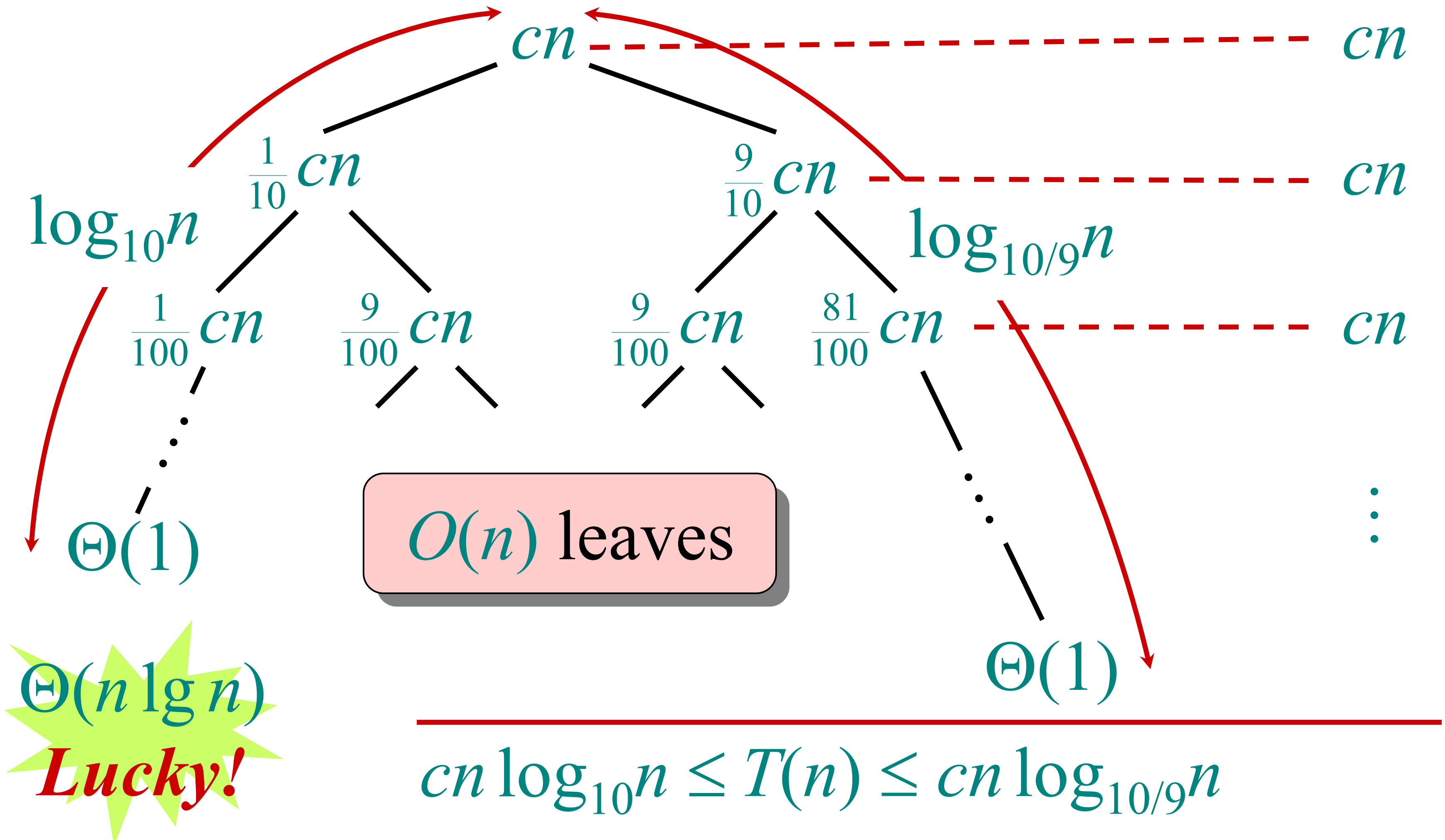
$$\frac{1}{10}cn \qquad \frac{9}{10}cn$$

$$T\left(\frac{1}{100}n\right) \; T\left(\frac{9}{100}n\right) \qquad T\left(\frac{9}{100}n\right) \; T\left(\frac{81}{100}n\right)$$

Analysis of "almost-best" case



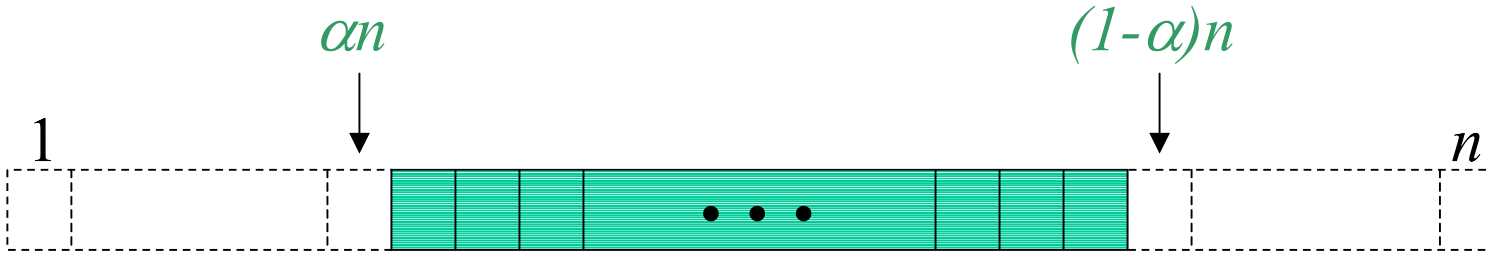
Analysis of “almost-best” case



$$cn \log_{10}n \le T(n) \le cn \log_{10/9}n$$

Balanced Partitionings: Splits of constant proportionality

- α–to–$(1-\alpha)$ proportional split yields $\Theta(n \lg n)$ time
- Let $\mathcal{P}_{\alpha>}$ = probability that H-PARTITION produces a split more balanced than an α–to–$(1-\alpha)$ split on a random array ($0< \alpha \leq 1/2$)
- P_q=probability that H-PARTITION returns q for any $1\leq q <n$
- $P_1=2/n$ and $P_q=1/n$ for $2\leq q <n$ for Hoare's partitioning algorithm



$$\triangleright\ \mathcal{P}_{\alpha>} = \sum_{q=\alpha n+1}^{(1-\alpha)n-1} P_q = \sum_{q=\alpha n+1}^{(1-\alpha)n-1} (1/n) = \frac{1}{n}\sum_{q=\alpha n+1}^{(1-\alpha)n-1} 1$$

$$= \frac{1}{n}\Big(((1-\alpha)n-1)-(\alpha n+1)+1\Big) = \frac{1}{n}\ ((1-\alpha)n-1-\alpha n-1+1)$$

$$= \frac{1}{n}(n-\alpha n-1-\alpha n) = \frac{1}{n}(n(1-2\alpha)-1)$$

$$\boxed{\mathcal{P}_{\alpha>} = (1-2\alpha) - 1/n \approx 1-2\alpha \text{ for large } n}$$

Balanced Partitionings

$$\mathcal{P}_{\alpha>} = 1-2\alpha$$

$\mathcal{P}_{0.1>} = 1-2\times0.1 = 0.80;$ even $\mathcal{P}_{0.01>} = 0.98$

- Hence, H-PARTITION produces a split
  - More balanced than a
    - 0.1–to–0.9 split %80 of the time
    - 0.01–to–0.99 split %98 of the time
  - Less balanced than a
    - 0.1–to–0.9 split %20 of the time
    - 0.01–to–0.99 split %2 of the time

Intuition for the average case

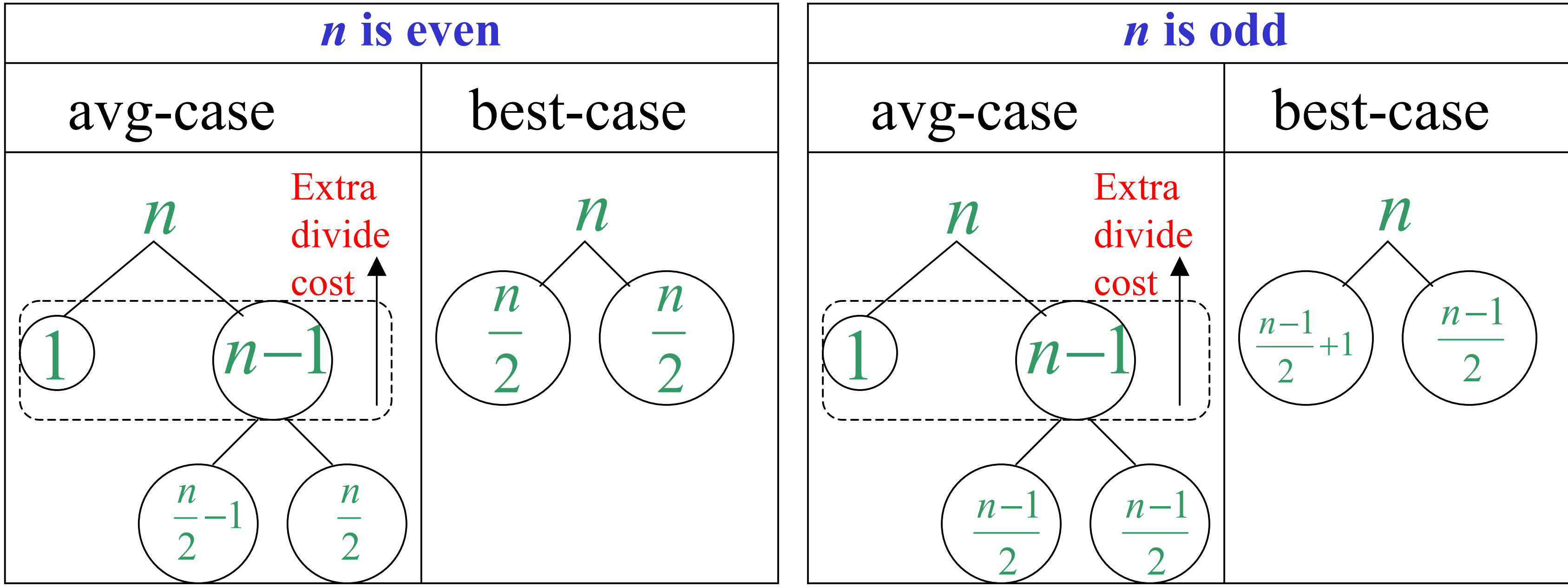
- Assumption: all permutations are equally likely
- Unlikely: splits always the same way at every level
- Expectation:
  - Some splits will be reasonably balanced
  - Some splits will be fairly unbalanced
- Average case: a mix of good and bad splits.

▷ *Good* and *bad* splits distributed randomly thru the tree

▷ Assume: *good* and *bad* splits occur in the alternate levels of the tree

  ▷ Good-Split: Best-case split, Bad-Split: Worst-case split

Intuition for the average case



- Two successive levels of avg-case produce a split
  - Slightly better than single level of best-case
  - Extra divide cost of $\Theta(1+(n-1)) = \Theta(n)$ at alternate levels
  - $\Theta(n)$ cost of bad splits absorbed into $\Theta(n)$ cost of good splits
- Running time is still $\Theta(n \lg n)$
  - But, slightly larger hidden constant
  - i.e. height of the tree $\approx$ twice of that of best-case

Intuition for the average case

Suppose we alternate lucky, unlucky, lucky, unlucky, lucky, ….

$$L(n) = 2U(n/2) + \Theta(n)$$ ***lucky*** ***(best)***

$$U(n) = L(n-1) + \Theta(n)$$ ***unlucky*** ***(worst)***

Solving:

$$\begin{aligned} L(n) &= 2(L(n/2-1) + \Theta(n/2)) + \Theta(n) \\ &= 2L(n/2-1) + \Theta(n) \\ &= \Theta(n \lg n) \end{aligned}$$

Lucky!

How can we make sure we are usually lucky?