Expected Runtime

Tufts University

Cheat of the day

$$\sum_{k=2}^{n-1} k \log k \leq \frac{1}{2} n^2 \log n - \frac{1}{8} n^2$$

Previously ...

Indicator Random Variables (IRV) is a new tool:

- Cheat of the **week**: $E[X + Y] = E[X] + E[Y]$,
- Works EVEN when X and Y are co-dependant
- **New**: $E[XY] = E[X]E[Y]$ if X and Y are independent

When computing expectancy of some complex event:

1. Define X (real goal) and X_i (single simple event)
2. Express X as a combination of X_i (ideally sums)
3. Compute $E[X_i]$
4. Compute $E[X]$ and use Cheat of the week
5. ???
6. Profit

Remember last lecture?

$\textsc{RandomAlgorithm}$

$x \leftarrow \textsc{RandomBit}$ (0 with 50% probability, 1 otherwise)

while $x \neq 0$

$x \leftarrow \textsc{RandomBit}$

Today's goal: learn how to analyze randomized algorithms

Using expectancy to analyze algorithms

$\textsc{RandomAlgorithm}$

$x \leftarrow \textsc{RandomBit}$ (0 with 50% probability, 1 otherwise)

while $x \neq 0$

$x \leftarrow \textsc{RandomBit}$

1. X = number of times the loop condition evaluates to true

   X_i = we loop at least i times
2. X = largest i such that $(X_i = 1) = \sum_{i=1}^{\infty} X_i$
3. $E[X_i] = P(X_i = 1) = \frac{1}{2^i}$
4. $E[X] = E[\sum_{i=1}^{\infty} X_i] = \sum_{i=1}^{\infty} E[X_i] = \sum_{i=1}^{\infty} \frac{1}{2^i} = 1$

Expected # of loops = 1 $\Rightarrow$ expected runtime is $O(1)$

Quicksort

$\textsc{Quicksort}(A, n)$

- If $n \leq 5$ sort A by brute force
- $i \leftarrow \textsc{RandomNumber}(n)$
- $\mathrm{pos} \leftarrow \textsc{Partition}(A, n, i)$
- $\textsc{Quicksort}(A[1 : \mathrm{pos} - 1], \mathrm{pos} - 1)$
- $\textsc{Quicksort}(A[\mathrm{pos} + 1 : n], n - \mathrm{pos})$

Runtime? Worst case? $\mathrm{pos} = 1$ or $\mathrm{pos} = n$

$$T(n) = \Theta(n) + T(n-1) \Rightarrow \Theta(n^2)$$

Runtime

- QUICKSORT worst case is $\Theta(n^2)$ but not fair
  - Even with good input, runtime is $\Theta(n^2)$
  - Needs extremely unlucky random choices
  - Probability(Always doing bad choice)= $\frac{2}{n} \cdot \frac{2}{n-1} \cdot \frac{2}{n-2} \dots \cdot \frac{2}{2} \approx \frac{1}{n^n}$

- What if we were bad but not terribly bad?

  Say, each pivot choice has rank $0.9n$
  $T(N) = \Theta(n) + T(0.1n) + T.(9n)$
  $\Rightarrow T(n) = \Theta(n \log n)$ (proof by substitution)
  What if rank is $0.999999n$? also $\Theta(n \log n)$

Expected Runtime

- Worst-case runtime of QUICKSORT is not accurate

  Depends on instance and random choices of algorithm

  User can control instance but NOT the random choices

- Let's talk about *average* instead

  PROBLEM: average depends on input

  is input sorted? Almost sorted? repeated numbers?

  Also depends on random choices. Do we pick good/bad pivot?

- Introducing *expected worst-case* runtime

  worst-case = worst possible instanc

  expected = averaged over random choices

Theorem (Today's goal)

QUICKSORT *runs in expected worst case* $\Theta(n \log n)$ *time*
(even for worst-case instance, averaged over its random choices*)*

Expected runtime analysis

Let $T(n)$ =runtime of worst-case QuickSort for n numbers

- $T(n) = \Theta(n) + T(i-1) + T(n-i)$ for some $i \leq n$

  Which one? Let's use IRVs to obtain a formal expression
- Let $X_i = 1$ if we selected i-th smallest number as pivot
- $T(n) = \Theta(n) + X_1(T(0) + T(n-1)) + X_2(T(1) + T(n-2)) + X_3 \ldots$
- Alternatively $T(n) = \Theta(n) + \sum_{i=1}^{n} X_i(T(i-1) + T(n-i))$
- Let's compute $E[T(n)]$

  $E[X_i] = \frac{1}{n}$

Runtime analysis

$E[T(n)] =$

$= E[\Theta(n)] + E[\sum_{i=1}^{n} X_i(T(i-1) + T(n-i))]$

$= \Theta(n) + \sum_{i=1}^{n} E[X_i(T(i-1) + T(n-i))]$ (lin of expectation)

$= \Theta(n) + \sum_{i=1}^{n} E[X_i]E[(T(i-1) + T(n-i))]$ (independent choice)

$= \Theta(n) + \sum_{i=1}^{n} \frac{1}{n}E[(T(i-1) + T(n-i))]$ (pivot chosen randomly)

$= \Theta(n) + \frac{1}{n}\sum_{i=1}^{n} E[T(i-1)] + E[T(n-i)]$ (more LoE)

$= \Theta(n) + \frac{1}{n}\sum_{i=1}^{n} E[T(i-1)] + \frac{1}{n}\sum_{i=1}^{n} E[T(n-i)]$ (algebra)

$= \Theta(n) + \frac{2}{n}\sum_{i=0}^{n-1} E[T(i)]$ (adding same terms twice)

$= \Theta(n) + \frac{2}{n}\sum_{i=2}^{n-1} E[T(i)]$ (minor technicality)

Runtime analysis

$E[T(n)] = \Theta(n) + \frac{2}{n}\sum_{i=2}^{n-1} E[T(i)]$

Claim: $E[T(n)] \leq cn \log n$

Proof by substitution

Base case: $E[T(2)] = \Theta(1) \leq d \leq c2log2 \rightarrow$ ok as long as $c \geq \frac{d}{2}$

Induction:

$$
\begin{aligned}
E[T(n)] &= \Theta(n) + \tfrac{2}{n}\textstyle\sum_{i=2}^{n-1} E[T(i)] \\
&\leq d'n + \tfrac{2}{n}\textstyle\sum_{i=2}^{n-1} ci \log i && \text{(induction hypothesis)} \\
&\leq d'n + \tfrac{2c}{n}\textstyle\sum_{i=2}^{n-1} i \log i && \text{(algebra)} \\
&= d'n + \tfrac{2c}{n}\left(\tfrac{1}{2}n^2 \log n - \tfrac{1}{8}n^2\right) && \text{(cheat of the day)} \\
&= d'n + cn \log n - \tfrac{c}{4}n && \text{(algebra)} \\
&= cn \log n - \left(\tfrac{c}{4} - d'\right)n && \text{(algebra)} \\
&\leq cn \log n && \text{(if } c \geq 4d'\text{)}
\end{aligned}
$$

Summary

- Analyzing probabilities with algorithms is complicated

  Cheat of the week to the rescue!

  We focus on expected (worst-case) runtime

- QuickSort is a great algorithm, but analysis is tough

  Input array has little impact on runtime

  Random coin tosses do

- More practice next lecture!