QuickSort

Data Structures and Algorithms

Andrei Bulatov

Probability Reminder

Sample space

Event

Probability

Discrete random variable:

A variable that takes values with certain probability

Example:

The amount of money you win buying a lottery ticket:

there are 1000 tickets, 1 wins \$10000, 10 win \$100, the rest win nothing

Pr[X = 10000] = 1/1000, Pr[X = 100] = 1/100, Pr[X = 0] = 989/1000

Random Variables

Expectation

Let X be a discrete random variable with values $v_1, \ldots, v_k$

Then $E[X] = v_1 \cdot \Pr[X = v_1] + \ldots + v_k \cdot \Pr[X = v_k]$

Example:

E[your win] = 10000·Pr[X = 10000] + 100·Pr[X = 100] + 0·Pr[X = 0]

= 10000· 1/1000 + 100 · 1/100 + 0·989/1000

= 11

One random variable interesting for us is the running time of some algorithm

Properties of Random Variables

Linearity: Let X , Y be discrete random variables, and α a number

Then

E[X + Y] = E[X] + E[Y]

E[αX] = α E[X]

Example:

We flip n fair coins. How many heads do we get on average?

$$X_i = \begin{cases} 1, & \text{if heads on } i\textit{th} \text{ flip} \\ 0, & \text{otherwise} \end{cases}$$

It is called an indicator variable

$$E[X_i] = 1 \cdot \Pr[X_i = 1] + 0 \cdot \Pr[X_i = 0]$$

Let $X = X_1 + \ldots + X_n$ be the total number of heads

$$E[X] = E[X_1 + \ldots + X_n] = E[X_1] + \ldots + E[X_n] = n \cdot \tfrac{1}{2} = \tfrac{n}{2}$$

Quicksort: Input Distribution

Inputs for Quicksort are permutations of numbers

Unrealistic Assumption:

All permutations are equiprobable

Then each of them appears with probability $\frac{1}{n!}$

QuickSort

```
QuickSort(A,p,r)
if p<r then do
   set q:=Partition(A,p,r)
   QuickSort(A,p,q-1)     Quicksort(A,q+1,r)
endif

Partition(A,p,r)
set x:=A[r],  set i:=p-1
for j=p to r-1 do
   if A[j]≤x then do
      set i:=i+1,    exchange A[i] and A[j]
   endif
endfor
exchange A[i+1] and A[r],  output i+1
```

Running time

Lemma

Let X be the number of comparisons performed in the if of the Partition procedure over the entire execution of Quicksort on an n-element array. Then the running time of Quicksort is $O(n + X)$.

Proof

Partition is called at most n times

Each of the calls does a constant amount of work and some number of iterations of the for loop.

During each iteration it does again a constant amount of work, including one comparison.

Therefore the total number of iterations of the for loop equals X, the number of comparisons

QED

Counting Comparisons

Lemma shows that it suffices to count the number of comparisons performed by the algorithm.

Let $z_1, z_2, \ldots, z_n$ be the numbers to sort such that z_i is the i*th* smallest element

Let $Z_{ij} = \{z_i, \ldots, z_j\}$

Observation

Every pair z_i, z_j is compared at most once

Indeed, every element can be a pivot at most once

Every comparison is performed with the current pivot

Counting Comparisons: Random Variables

Let X_{ij} be the number of times z_i is compared with z_j during the execution of the algorithm

Then $X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$

We are interested in the average value of X, that is its expectation

$$E[X] = E\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}\right]$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}]$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Pr[z_i \text{ is compared to } z_j]$$

Finding Probability

Consider z_i and z_j

If the first element x chosen from Z_{ij} as pivot $z_i < x < z_j$ then z_i and z_j are never compared

If z_i or z_j is chosen first, then it is compared to the other element

Pr[z_i is compared to z_j] = Pr[z_i or z_j is first pivot chosen from Z_{ij}]

= Pr[z_i is first pivot chosen from Z_{ij}]

+ Pr[z_j is first pivot chosen from Z_{ij}]

$$= \frac{2}{j-i+1}$$

Finding Expectation

Now we can use it to find the expected time

$$
\begin{aligned}
E[X] &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Pr[z_i \text{ is compared to } z_j] \\
&= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \\
&= \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1} \\
&< \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k} \\
&< \sum_{i=1}^{n-1} \log n = O(n \log n)
\end{aligned}
$$

Quicksort: Running Time

Theorem

The expected running time of Quicksort is in $O(n \log n)$

Homework

Show that the running time of QuickSort is $\Theta(n^2)$ when the array A contains distinct elements and is sorted in decreasing order

What is the running time of QuickSort when all elements of array A have the same value?