Lecture #22: Application : Analysis of Algorithms
Lecture Presentation

Analyzing a Randomized Algorithm

The lecture presentation included a ***randomized algorithm*** to check whether an input integer `key` was stored in an input integer array `A` — as given in Figure 2 on page 4. This algorithm calls a variant of a "Linear Search" algorithm as a subroutine — as shown in Figure 1 on page 2.

The goal for this initial problem is to analyze the ***running time*** of this randomized algorithm when its input includes an integer array `A` with length n.

Number of Steps Used by the Deterministic Subroutine:

```
integer dSearch (integer[] A, integer key) {
  integer n := A.length
  integer i := 0
  while (i < n) {
    if (A[i] == key) {
      return true
    }
    i := i + 1
  }
  return false
}
```

Figure 1: Subroutine Implementing Linear Search

```
boolean rSearch3 (integer[] A, integer key) {
1. integer n := A.length
2. integer i := 0
3. while (i < n) {
4.   Choose j uniformly from the set {0, 1, 2, ..., n − 1} — independently from any
     previous selections.
5.   if (A[j] == key) {
6.     return true
     }
7.   i := i + 1
   }
8. return dSearch(A, key)
}
```

Figure 2: Randomized Algorithm for Searching in an Integer Array

Successful Search:

Let k be an integer such that $1 \leq k \leq n$ and suppose that exactly k copies of the input `key` are stored in the array `A`. Suppose, as well, that the *first* copy of the `key` is stored at index ℓ — so that ℓ is an integer such that $0 \leq \ell \leq n - k$.

Defining the Sample Space

Defining the Probability Distribution

Defining the Random Variable

Completing the Analysis for This Input

Unsuccessful Search:

Suppose, instead, that there are no copies of the `key` in the input array, at all.

What about the Sample Space, etc.?

Worst-Case Expected Running Time:

1 Dealing with a Complicated Expression

This kind of analysis can result in complicated expressions, whose value we wish to give in "closed form". If we cannot give a simplified version of the expression's value then we might be able to work with a simplified form of an ***approximation***, ***upper bound*** or ***lower bound*** for this expression, instead.

One such expression was found in the preparatory reading — where the expected value of a random variable was found to be

$$\mathsf{E}[T] = \sum_{i=0}^{n-1} \left(1 - \tfrac{k}{n}\right)^i \cdot \tfrac{k}{n} \cdot (4i+6) + \left(1 - \tfrac{k}{n}\right)^n \cdot (4n+3).$$

It is doubtful that this kind of expression will have a simple solution — but it is possible to approximate its value.

An Easy Case: What is this when $k = 0$?

Another Easy Case: What is this when $k = n$?

Now suppose, instead, that k is an integer such that $1 \leq k \leq n - 1$.

Making a Start:

Now, what can we do with the expression $\left(1 - \frac{k}{n}\right)^n$?