Randomized Algorithms

Spørgsmål 4 fra Exam Questions

Kevin Vinther

December 11, 2023

Table of Contents i

Randomly Permuting Arrays

Randomly Permuting Arrays

- Givet et array, vil vi gerne have en måde hvorpå vi kan permutere det tilfdæligt.
- Der er flere måder at gøre dette på.
- En af måderne:
  - Giv hvert element i array A[i] en prioritet P[i], og sortér after prioriteterne.

Permute-By-Sorting

$n \leftarrow A$.length
Let $P[1..n]$ be a new array
for $i \leftarrow 1$ to n **do**
    $P[i] \leftarrow \text{RANDOM}(1, n^3)$
end for
sort A, using P as sort keys

Lemma 5.4

Lemma (Lemma 5.4)

Procedure `Permute-By-Sorting` *produces a uniform random permutation of the input, assuming that all priorities are distinct*

Lemma 5.4 Bevis i

Vi starter med at overveje den præcise permutation hvor elementet $A[i]$ får den i'e laveste prioritet. Vi viser at denne permutation sker med den præcise sandsynlighed $1/n!, \mathit{for} i =, 1, 2, \ldots, n$. Lad E_i være hændelsen at elementet A_i får den $i'e$ mindste prioritet. Vi ønsker at udregne sandsynligheden for alle i.

$$P(E_1 \cap E_2 \cap \cdots \cap E_{n-1} \cap E_n)$$

Denne sandsynlighed er lig med (iflg. opgave C.2-5):

$$P(E_1) \cdot P(E_2 | E_1) \cdots P(E_i | E_{i-1} \cap E_{i-2} \cap \cdots \cap E_1) \cdots P(E_n | E_{n-1} \cap \cdots \cap E_1)$$

Lemma 5.4 Bevis ii

Vi har at $p(E_1) = 1/n = 1/n = 1/n = 1/n$ fordi det er sandsynligheden at en prioritet valgt tildældigt ud af et sæt af n er den mindste sandsynlighed.

Som det næste observerer vi at $P(E_2|E_1) = \frac{1}{n-1}$ fordi, givet at elementet $A[1]$ har den mindste prioritet, har hver af de resterende $n-1$ elementer en lige chance for at have den anden mindste prioritet. Generelt:

$$p(E_i|E_{i-1} \cap E_{i-2} \cap \cdots \cap E_1) = \frac{1}{(n-i+1)}$$

Dermed har vi:

Lemma 5.4 Bevis iii

$$p(E_1 \cap E_2 \cap E_3 \cap \cdots \cap E_{n-1} \cap E_n) = \left(\frac{1}{n}\right)\left(\frac{1}{n-1}\right)\cdots\left(\frac{1}{2}\right)\left(\frac{1}{1}\right) = \frac{1}{n!}$$

Randomize-in-Place

- Vi kan gøre det endnu bedre. Permute-By-Sorting var $\Theta(n \lg n)$. Randomize-In-Place er $O(n)$

Randomize-in-Place

$n \leftarrow A$.length
for $i \leftarrow 1$ to n **do**
    swap $A[i]$ with $A[Random(i, n)]$
end for

Lemma 5.5

Lemma (Lemma 5.5)

Procedure `Randomize-in-Place` computes a uniform random permutation.

Lemma 5.5 Bevis i

Vi bruger den følgende loop invariant[1]: Lige før den $i'e$ iteration af **for** loopet (linjerne 2-3), for hver mulig $(i-1)$-permutation af de n elementer; indeholder subarrayen $A[1..i-1]$ denne $(i-1)$-permutation med sandsynligheden $(n-i+1)!/n!$

(Altså, sandsynligheden for at subarrayen indtil videre er en præcis subarray skal gerne være $(n-i+1)!/n!$ (for ellers er den ikke uniformly random.))

Vi skal så vise at invarianten er sand:

- Før den første loop
- I hvert loop efterfølgende
- Når loopet er færdigt

[1] En invariant skal være sand under hele beviset