PROBABILISTISKA ALGORITMER

ALGORITMER SOM ANVÄNDER SLUMP.

EXEMPEL: QUICKSORT MED SLUMP

```
RandomQuicksort(v[i..j]) =
  IF i<j THEN
    x ← v[RANDOM(i,j)]
    m ← PARTITION(v[i..j],x)
    RandomQuicksort(v[i..m-1])
    RandomQuicksort(v[m+1..j])
```

ANALYS: (BEVISET SKIPPAT)

MED SANNOLIKHET $> 1-\frac{1}{n^k}$ TAR RANDOMQUICKSORT TID $O(n \log n)$, FÖR VARJE KONSTANT k.

LAS VEGAS-ALGORITM — PROBABILISTISK ALGORITM SOM ALLTID SVARAR RÄTT, MEN KAN TA OLIKA LÅNG TID PÅ SEJ.

EXEMPEL: MAX 3CNFSAT

GIVET EN MÄNGD $\{c_i\}_{i=1}^{n}$ MED KLAUSULER ÖVER BOOLESKA VARIABLER $x_1, \ldots, x_m$, HITTA EN VARIABELTILLDELNING SOM SATISFIERAR SÅ MÅNGA KLAUSULER SOM MÖJLIGT.

```
RandomMax3CNFSAT({ci}) =
  FOR i←1 TO m DO
    xi ← RANDOM(0,1)
  RETURN {xi}
```

ANALYS:

TID: $O(n)$

$P[c_i \text{ INTE SATISFIERAD}] = \frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2} = \frac{1}{8}$

$Y =$ ANTAL SATISFIERADE KLAUSULER.

$E[Y] = n \cdot P[c_i \text{ SATISFIERAD}] = n\left(1-\frac{1}{8}\right) = \frac{7}{8}n$

$\Rightarrow$ APPROXIMERAR MAX3CNFSAT INOM $\frac{8}{7}$.

MONTE CARLO-ALGORITM — PROBABILISTISK ALGORITM SOM ALLTID GÅR SNABBT, MEN SOM BARA IBLAND GER BRA RESULTAT.

SLUMPELIMINERING

GÖR EN PROBABILISTISK ALGORITM DETERMINISTISK GENOM ATT ARBETA BORT SLUMPEN.

METOD 1: PRÖVA ALLA UTFALL

OM k SLUMPBITAR BEHÖVS I EN ALGORITM FINNS DET 2^k MÖJLIGA UTFALL. TESTA ALLA DESSA OCH PLOCKA DET BÄSTA!

PROBLEM: TAR LÅNG TID OM k ÄR STORT.

METOD 2: BETINGADE VÄNTEVÄRDEN

BERÄKNA $E[Y \mid x_1=0]$ OCH $E[Y \mid x_1=1]$

VÄLJ DET VÄRDE PÅ x_1 SOM GER STÖRST BETINGAT VÄNTEVÄRDE.
FORTSÄTT PÅ SAMMA SÄTT MED RESTERANDE x_i.

PROBLEM: BETINGADE VÄNTEVÄRDENA KAN VARA SVÅRA ATT BERÄKNA.

FLER METODER FINNS!

> GÅR DET ALLTID ATT SLUMPELIMINERA EN PROBABILISTISK POLYNOMISK ALGORITM TILL EN DETERMINISTISK POLYNOMISK ALGORITM?