

IMPA Omgång 3 – 2010-09-22

10706 – Number Sequence

Här finns säkert många lösningar. En lösning är att skapa en funktion som räknar ut längden på texten av sekvensen 1..k. Detta kan göras med en for-loop som varje varv multiplicerar 10 till loop-variabeln. Hur många sådana sekvenser behövs? (Loopa enkelt.) När man vet detta kan man räkna baklänges för att hitta var i sista sekvensen som man ska plocka tecknet ifrån.

10987 – AntiFloyd

Floyd-Warshall visar vilka vägar som inte ska användas, dvs. om $\text{dist}[i][k] > \text{dist}[i][j] + \text{dist}[j][k]$ så är vägen mellan i och k inte giltig. Vid likhet behövs inte vägen.

11560 – Fixing the Bugs

Svår dynamisk programmering. Betrakta följande frågor:

Givet sannolikheten för att lösa varje problem just nu, vilket ska jag välja? Denna fråga kan besvaras direkt med en girig algoritm ty det är alltid bäst att prova det som har högst väntevärde.

```
bug = GetNextBug(bits bugsLeft, double[] probabilities)
```

Givet vilka buggar som är kvar och hur mycket tid man har använd på dem, hur stor är sannolikheten att man löser en av dem?

```
[probs, timeOnBugs, bestNextBug] =  
    AllProbs(bits bugsLeft, int[] timeUsedOnTheseBugs)  
    // Returns information for every bug
```

```
[probs, timeOnBugs, bestNextBug] =  
    AllProbs(bugsLeft, timeUsedOnTheseBugs - 1);  
bestNextBug = GetNextBug(bugsLeft, probs);  
probs(bestNextBug) *= discount;  
timeOnBugs(bestNextBug)++;
```

Antal kombinationer av indata är $2^{10} \cdot 100 = 10^5$. Spara resultat med memoisation eller DP.

Givet hur lång tid som är kvar, vilka buggar som är kvar och tid som har använts på de kvarvarande buggarna, vad är väntevärdet för lösta buggar?

```
expected = Solved(int timeLeft, bits bugsLeft,  
    int timeUsedOnTheseBugs)
```

```
[probs, timeOnBugs, bug] =  
    AllProbs(bugsLeft, timeUsedOnTheseBugs - 1);  
prob= probs(bug);  
timeOnThisBug = timeOnBugs(bug);  
  
expected = prob * (Solved(timeLeft-1, bugsLeft & ~bit(bug),  
    timeUsedOnTheseBugs - timeOnThisBug) +  
    severity(bug)); // If solved  
expected += (1-prob) * Solved(timeLeft-1, bugsLeft,  
    timeUsedOnTheseBugs + 1); // If not solved
```

Med $100 \cdot 2^{10} \cdot 100 = 10^7$ indatakombinationer bör resultatet sparas med memoisation eller DP.

Ja, all returdata till memoisation får plats i minnet.