IV003 – sada 2, príklad 3

Vypracovali: Henrich Lauko 410438
Karel Kubíček 408351

Celkovú pravdepodobnosť, že nám padne k orlov spočítame ako súčet všetkých prípadov padnutia k orlov v n hodoch. Každý z týchto prípadov môžeme definovať ako súčin k pravdepodobností, že nám padne orol a súčin $n-k$ pravděpodobností, že nám padne panna. Náš algoritmus vychádza z rekurzívneho vzťahu, kde $P(k,n)$ vyjadruje pravdepodobnosť, že padne k orlov na n minciach:

$$P(k,n) = \begin{cases} 1 & \text{ak } k = 0 \text{ a } n = 0 \\ 0 & \text{ak } n = 0 \text{ a } k \text{ je ľubovolné} \\ P(k,n-1)\cdot(1-p_n) + P(k-1,n-1)\cdot p_n & \text{inak} \end{cases}$$

Pre dynamický návrh použijeme metódu memoizácie, kde si budeme predpočítavať predchádzjúce postupnosti hodov. Teda pre výpočet budeme potrebovať maticu M veľkosti $(n-k+1)\times k+1$ kde v bode $M[0][0]$ je pravdepodobnosť 1, že na 0 hodov padne 0 orlov. V $M[n-k][k] = P(k \text{ orlov na } n \text{ hodov})$. Algoritmus začne v bode $[0][0]$ a bude predpočítavať hodnoty až ku bodu $[n-k][k]$ nasledovne:

1. prípad predpočítame si hodnotu, že sme hodili len t orlov a žiadne panny, čo je súčin pravdepodobností $\prod_{i=1}^{t} p_i$, ktorý si pri každom kroku vieme predpočítať z predchadzajúceho výsledku ako $(\prod_{i=1}^{t-1} p_i)\cdot p_t$.

2. prípad je zhodný s prvým len počítame koľko sme hodili panien namiesto orlov, teda berieme pravdepodobnosť že nám pri i-tom pokuse padne panna ako $(1-p_i)$

3. prípad vychádza zo znalosti, že vieme pravdepodobnosť s akou padne t orlov a $h-1$ panien, a $t-1$ orlov a h panien, kde z počiatočnej rekurzívnej rovnice vieme, že $M[h][t] = M[h-1][t]\cdot(1-p_t) + M[h][t-1]\cdot p_t$.

Algoritmus 1: PROBABILITY(n,k,P)

vstup: n je počet mincí, k počet chcených orlov, P pole pravděpodobností, kde $P[i]$ vyjadruje, že na i-tej minci padne orol
výstup: pravdepodobnost, že padne k orlů na n minciach

$M[0][0] \leftarrow 1$ // *pravdepodobnosť, že na 0 hodov padne 0 orlov*
for $h \leftarrow 0$ **to** $n-k$ **do**
  for $t \leftarrow 0$ **to** k **do**
    if $t \neq 0 \wedge h = 0$ **then**
      $M[h][t] \leftarrow M[h][t-1]\cdot P[t]$ // *prípad 1*
    else if $h \neq 0 \wedge t = 0$ **then**
      $M[h][t] \leftarrow M[h-1][t]\cdot(1-P[h+1])$ // *prípad 2*
    else
      $M[h][t] \leftarrow M[h-1][t]\cdot(1-P[t+1]) + M[h][t-1]\cdot P[t]$ // *prípad 3*
  od
od
return $M[n-k][k]$

IV003 – sada 2, príklad 3

Vypracovali: Henrich Lauko 410438
Karel Kubíček 408351

Časová složitost Algoritmus počas svojho výpočtu vypĺňa maticu veľkosti $(n-k+1) \times k+1$, kde v najhoršom prípade $k = n/2$ a teda musíme spočítať $n/2 \cdot n/2$ hodnôt $\in \mathcal{O}(n^2)$. Keďže operácie v cykloch dokážeme robiť v konštantnom čase, celková časová zložitosť je $\mathcal{O}(n^2)$.

Konečnosť V cykloch algoritmu rastú indexy vždy o 1, vzhľadom k tomu, že indexy v tele cyklu nemeníme z vlastností for-cyklu vieme, že algoritmus,vykoná v cykloch len $(n-k) \times k$ iterácií, teda aj skončí.

Korektnosť Aby bol algoritmus korektný musí byť konečný a parciálne korektný. Konečnosť sme si už potvrdili pre parciálnu korektnosť dokážeme, že vo výslednom bode matice bude súčet všetkých možných hodov k orlov. Pre potreby dôkazu si môžeme maticu M predstaviť ako orientovaný graf(smer je od menších indexov matice k väčším), kde jednotlivé hodnoty matice predstavujú uzly, ktoré su spojené len so susednými hodnotami v horizontále a vertikále. Každý tento prechod teda definuje hod jednou mincou(vertikálny prechod - padla panna a horizontálny prechod - padol orol). My chceme ukázať, že všetky cesty dĺžky n vychádzajúce z bodu [0][0] do bodu $[n-k][k]$ tvoria všetky možné permutácie hodov $n-k$ panien a k orlov. Z vzdialenosti v tomto grafe vieme, že najkratšia cesta medzi bodom [0][0] a bodom $[n-k][k]$ je práve n. Teda nedokážeme túto cestu prejsť ináč ako na $[n-k]$ prechodov kde hodíme pannu a k prechodov kde hodíme orla. V každom uzle sa rozhodujeme či pôjdeme po prechode orla alebo panny, keďže týchto rozhodnutí je $n$neexistuje postúpnosť orlov a panien pre ktorú by neexistovala postúpnosť rozhodnutí v tomto grafe a obsahovala by k orlov a $n-k$ panien. Z algoritmu vidíme, že všetky tieto cesty sa postupne sčítajú, teda je parcíalne korektný a aj korektný.