



Maya extension

Extension Maya

Maya-Kalenderreform

Christian Kauth

Hard Task — Emergency

Heidi messed things up and got lost inside the pyramid of Figure 1. Oops ! She might actually stand in any corridor with equal probability, facing any direction. As Heidi brought some high-tech equipment from the present times, you may simply broadcast emergency directions to Heidi which will bring her out of the pyramid as soon as possible, with or without having updated the calendar ! Heidi will follow your instructions according to her previous strategy but cannot provide you with any feedback. The only feedback you may get is herself appearing at the exit of the pyramid ! Once Heidi is out, she will obviously not return into that maze again, even if you continue giving her instructions. Note that counting happens cyclically, and counting n ($n > 0$) entrances in room i will correspond to the $n \pmod{c_i}$ -th exit to the right.

You are to write a function `void get_heidi_out()` that

- Sends instructions to Heidi on turns to take via calls to the function `void take_exit_number(int n)`.
- Chooses those calls such that we can statistically expect Heidi out of the pyramid of figure 1 as soon as possible. By *as soon as possible* we mean a sequence that has minimal expectation on entrance counts (just as in subtask 2, minimize the entrances counted, not the number of rooms visited/corridors traversed).

Tâche Difficile — Urgence

Heidi s'est plantée et a fini par se perdre dans la pyramide de la Figure 1. Oups ! Elle peut se trouver dans n'importe quel couloir avec la même probabilité, regardant dans l'une ou l'autre direction. Heureusement, Heidi a pris avec elle de l'équipement hi-tech de nos jours, vous pouvez donc lui transmettre des informations sur le chemin à suivre pour sortir de la pyramide le plus vite possible, peu importe la mise à jour du calendrier ! Heidi suivra vos instructions selon la stratégie précédente mais elle ne pourra vous donner aucun feedback. Le seul feedback que vous aurez sera quand elle apparaîtra à la sortie de la pyramide ! Lorsque Heidi sera sortie, elle ne retournera évidemment pas dans ce labyrinthe une fois de plus, même si vous deviez continuer à lui donner des instructions. Notez que Heidi compte de manière cyclique : Si elle compte n ($n > 0$) entrées dans la salle i , elle prendra la $n \pmod{c_i}$ -ième sortie à sa droite.

Vous devez écrire la fonction `void get_heidi_out()` qui

- Envoie des instructions à Heidi sur les couloirs à prendre via la fonction `void take_exit_number(int n)`.
- Choisit les appels de sorte à ce qu'on puisse statistiquement s'attendre à ce que Heidi sorte de la pyramide de la figure 1 au plus vite. Par *au plus vite* on entend la séquence qui minimise l'espérance mathématique du nombre d'entrées comptées (tout comme dans la tâche moyenne, minimisez les entrées comptées, non pas le nombre de salles/couloirs traversés).

Schwierige Aufgabe — Notfall

Heidi hat sich in der Pyramide der Abbildung 1 verloren. Uuups ! Sie könnte sich in jedem Korridor mit gleicher Wahrscheinlichkeit befinden, in irgendeine Richtung spazierend. Dank ihrem Handy aus heutiger Zeit könnt ihr Heidi einfach Notfallrichtungen zufunkeln, welche sie so schnell wie möglich aus der Pyramide herausführen. Ob sie dabei den Kalender findet oder nicht, ist egal ! Heidi führt eure Befehle gemäss der vorherigen Strategie aus, kann euch aber kein Feedback geben. Euer einziges Feedback ist, wenn Heidi am Pyramidenausgang erscheint ! Einmal raus, wird sie natürlich nicht mehr ins Labyrinth zurückkehren, selbst wenn ihr weitere Befehle erteilt. Beachtet, dass Heidi zyklisch zählt. Befehlt ihr Heidi zum Beispiel, den n -ten ($n > 0$) Ausgang zu nehmen, wenn sie gerade in Raum i eintritt, nimmt sie in der Tat den $n \pmod{c_i}$ -ten zu ihrer Rechten.

Schreibt dazu eine Funktion `void get_heidi_out()` welche

- Heidi Drehbefehle durch Aufrufe der Funktion `void take_exit_number(int n)` zusendet.
- Diese Aufrufe so wählt, dass ihr Heidi statistisch so bald wie möglich aus der Pyramide der Figur 1 herausführt. Unter *so bald wie möglich* verstehen wir eine Sequenz, welche einen minimalen Erwartungswert an gezählten Tunnelleingängen hat (genau wie in Unteraufgabe 2, sollt ihr die gezählten Eingänge minimieren, nicht die Anzahl der durquerten Räume oder Korridore).