



# Cyberfood

Christian Kauth

## Hard Task

Heidi has a sweet tooth and likes occasionally grazing leaves from tries, radix, suffix, interval and other trees. She realized that different trees yield different sets of random tastes and each tree type has probability  $p_i$  to yield a leaf flavour that Heidi likes. The  $p_i$  derive from a uniform distribution over  $[0, 1]$ . Given that there are exactly 10 trees per forest and that a full meal consists of 100 leaves, instruct Heidi from which tree she shall grab a leaf. She will let you know then whether she liked it or not. Write a function `void happy_meal()` that calls exactly 100 times `bool eat(int id)` to eat a leaf from tree  $1 \leq id \leq 10$ . The function returns `true` if Heidi liked it, `false` otherwise. There will be  $10^4$  calls to `happy_meal` with different forests. Make her meals as enjoyable as you can! You pass the tests if overall, at least 90% of all leaves are as tasty as you could make them if you knew the  $p_i$ s in advance.

## Tâche Difficile

Heidi a le bec sucré et aime goûter de temps en temps les feuilles de tries, d'arbres radix, à intervalles, de suffixes, et d'autres arbres. Elle s'est rendue compte que différents arbres produisent différents ensembles aléatoires de goûts, et que pour un type d'arbre donné, chaque feuille a une probabilité  $p_i$  (distribuée uniformément entre 0 et 1) d'avoir un goût qui plaît à Heidi. Étant donné qu'il y a exactement 10 arbres par forêt, et qu'un repas complet se compose de 100 feuilles, dites à Heidi de quelle arbre elle doit prendre la prochaine feuille. Elle vous dira si la feuille lui a plu ou non. Écrivez une fonction `void happy_meal()` qui appelle exactement 100 fois la fonction `bool eat(int id)`, qui indique que Heidi doit manger la prochaine feuille de l'arbre  $1 \leq id \leq 10$ . Cette fonction retourne `true` si Heidi a aimé la feuille, et `false` sinon. Il y a  $10^4$  forêts différentes à explorer, et nous appellerons `happy_meal()` pour chacune. Faites en sorte que Heidi mange le plus grand nombre de feuilles qui lui plaisent. Cette tâche sera considérée comme résolue si au moins 90% de toutes les feuilles sont aussi savoureuses que dans la solution optimale (où vous savez les  $p_i$  à l'avance).

## Schwierige Aufgabe

Heidi ist ein ziemliches Schleckmaul. Sie probiert gerne mal Blätter von Tries, Radix-, Suffix-, Intervall- und anderen Bäumen. Dabei fand sie heraus, dass jeder Baum ganz unterschiedlich schmackhafte Blätter hat. Für jeden Baum gibt es eine Wahrscheinlichkeit  $p_i$ , dass Heidi ein Blatt dieses Baumes mag. Diese  $p_i$  sind gleichmässig zufällig zwischen  $[0, 1]$  verteilt. Ein Wald im Cyberspace besteht aus 10 Bäumen, und eine volle Mahlzeit enthält 100 Blätter. Helft Heidi, die richtigen Bäume zum Probieren auszuwählen, so dass ihre Mahlzeit so fein wie möglich ist. Ihr bestimmt, was Heidi probiert, und sie sagt euch, obs fein war oder nicht.

Schreibt eine Funktion `void happy_meal()`. Diese soll genau 100 Mal `bool eat(int id)` aufrufen, um ein Blatt von Baum  $1 \leq id \leq 10$  zu probieren. Die Funktion gibt `true` zurück, wenn Heidi das Blatt mochte, sonst `false`. Der Testserver wird eure Funktion `happy_meal`  $10^4$  Mal für verschiedene Wälder aufrufen. Euer Programm besteht, wenn insgesamt mindestens 90% der Blätter so fein sind, wie sie sein könnten, wenn ihr all die  $p_i$  im Voraus gekannt hättet.