

Soluciones entrenos OIE

Memory- IOI 2010

Un juego llamado Memory se juega con 50 cartas. Cada carta tiene una de las letras de la A a la Y (ASCII 65 a 89) impresa en su cara no visible, de modo que cada letra aparece exactamente en dos tarjetas. Las cartas se barajan en un orden aleatorio y se reparten boca abajo sobre la mesa. Jack juega al juego poniendo dos cartas boca arriba para que las letras sean visibles. Por cada una de las 25 letras, Jack recibe un caramelo la primera vez que ve ambas copias de la letra en las dos cartas boca arriba a la vez. Por ejemplo, la primera vez que Jack da la vuelta a las dos cartas que contienen la letra M, recibe un caramelo. Independientemente de si las letras son iguales o no, Jack luego coloca ambas cartas boca abajo de nuevo. El proceso se repite hasta que Jack recibe 25 caramelos, uno por cada pareja con la misma letra. Debes implementar una función `play` que juegue la partida. Tu implementación debe llamar a la función `faceup (C)`, que ya está implementada como parte del grader. `C` es un número entre 1 y 50 que indica una carta en particular que quieres colocar boca arriba. La carta no debe estar boca arriba en el momento de la llamada. `faceup (C)` devuelve la letra que está impresa en la carta `C`. Las llamadas a `faceup (C)` van por pares. La primera llamada levanta una carta y la deja boca arriba. La segunda llamada levanta otra carta y, tras procesar su valor en relación a la primera carta, vuelve a colocar ambas boca abajo. Tu función `play` solo puede terminar una vez que Jack haya recibido los 25 caramelos. Está permitido hacer llamadas a `faceup (C)` incluso después del momento en el que Jack reciba el último caramelo. Tu puntuación se calculará en función del número de llamadas a `faceup (C)` por parte de `play`.

Constraints

- 50 Puntos: cualquier estrategia que siga las reglas del juego y termine dentro del límite de tiempo.
- 100 Puntos: cualquier estrategia que siga las reglas del juego y termine dentro del límite de tiempo, haciendo a lo sumo 100 llamadas a `faceup (C)`.

Solución

Para conseguir los 100 puntos, basta con una solución que no levante una carta más de 100 veces. Esto significa que cada carta se puede levantar hasta dos veces, lo que nos permite resolver el problema con una función que hace dos pases por las cartas. Primero, levantamos cada carta una vez, guardando la letra que sale. Para saber qué dos cartas tienen cada letra, tenemos dos arrays, uno con una de las posiciones que tiene la carta y otro con la suma de las dos posiciones, de la cual podemos obtener mediante una simple resta la posición de la segunda carta. De esta forma, nos ahorramos el comprobar si ya se ha encontrado esa letra cada vez que levantemos una carta. Después, hacemos otro pase por las cartas, levantando juntas las que sabemos que tienen la misma letra debajo. Así, en 100 llamadas exactas a `faceup (C)` aseguramos el resultado.

Código

C++

```

1 void play() {
2     int pos[25]; // pos[i] contiene una de las posiciones de esa letra
3     int sumPos[25] = {0}; // sumPos[i] contiene la suma de las 2 posiciones de
4     ↪ esa letra
5     int curChar;
6     for (int i = 1; i <= 50; i++) {
7         // levantamos todas las cartas y guardamos el resultado
8         curChar = faceUp(i) - 'a';
9         pos[curChar] = i;
10        sumPos[curChar] += i;
11    }
12    for (int i = 0; i < 25; i++) {
13        // levantamos las parejas de cartas que contienen la misma letra
14        faceUp(pos[i]);
15        faceUp(sumPos[i] - pos[i]);
16    }
17 }

```