



XIFRES I LLETRES

Projecte final Algorísmia

UIB

Aplicació que permet jugar al joc televisiu conegut com “Xifres i Lletres”

Ahmed Antonio Boutaour Sanchez

43202227Q

Bernat Pericàs Serra

43212796M

Manual d'usuari

La interfície és trivial i quasi idèntica pels dos jocs, encara així es descriurà el seu funcionament.

Xifres

Per a generar els nombres inicials aleatòriament hi ha dos botons, un per a generar 7 nombres i mantenir el temps de càlcul baix, i l'altre per a generar fins a 10 nombres, augmentant considerablement el temps de càlcul, posant a prova les capacitats del Netbeans i de l'ordenador en general. A continuació si es clica a Genera solució, es mostrarà el procés per arribar a la millor solució. El requadre de baix està disponible per a la introducció de la resposta per part de l'usuari però ha de complir certs requisits: les operacions han d'estar en parèntesis i totes en una línia, fora afegir el resultat final ni cap "=". Finalment el botó de neteja buida el requadre per a introduir text i el botó de sortida tanca el programa.

Lletres

Per a les lletres, només hi ha un botó de generar, que genera 8 lletres per defecte, on són com a mínim 30% vocals i 40% consonants. Després al espitjar el botó de generar solució, es llistaran totes les paraules de longitud major a 5 formades per les lletres inicials i que estiguin contingudes al diccionari. A continuació trobam el botó de comprovar, que comprova si les paraules escrites al requadre són solucions correctes

Cost de complexitat

Xifres

L'algorisme encarregat de calcular les solucions del joc de Xifres es correspon amb els següents paràmetres:

$$a = 4$$

$$c = 1$$

$$\alpha = 0$$

I aplicant el Teorema per resta, com que $(a = 4) > (c^0 = 1)$, el cost de l'algorisme serà:

$$T(n) \in O((a^{1/c})^n) = 4^{1 \cdot n} = 4^n$$

Lletres

Per altra banda, l'algorisme del joc de les Lletres es correspon amb:

$$a = 1$$

$$c = 1$$

$$\alpha = 1$$

I aplicant un altre cop el Teorema per divisió, ja que $(a = 1) < (c^0 = 1)$, el cost serà:

$$T(n) \in O(n^{\alpha+1}) = n^2$$

Proves unitàries realitzades

Xifres

S'han realitzat les següents proves unitàries:

transformacio1: comprova si {0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 452910} es transforma en [2,3,4,7,8,75,100]

transformacio2: comprova si {1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 452910} es transforma en [1,2,3,4,7,8,9,75]

calcul1: comprova si la suma $1+1$ dona 2

calcul2: comprova si la resta $10-9$ dona 1

calcul3: comprova si la resta $10-19$ dona 0 (negatiu)

calcul4: comprova si la divisió $10/1$ dona 0 (operació podada)

backtracking1: comprova que el resultat més òptim en aquest cas és $2*3=6$

backtracking2 :comprova que el resultat més òptim en aquest cas és $2*5=10$

backtracking3 : comprova que el resultat no és l'esperat, $2*5 \neq 14$

solucionar1 : comprova que el resultat és 1, perquè es realitza d'acord a les regles establertes.

solucionar2 : comprova en aquest darrer cas que no és el resultat esperat, retornant un 0.

Estratègies seguides

Xifres

Per a trobar les solucions de les xifres s'ha seguit la estratègia del backtracking. Com a podes s'han eliminat les branques on la resta dona negatiu, si es multiplica per u, i finalment si es divideix entre u. A més, mb l'aplicació del doble for hem aprofitat 1/2/2017 la capacitat commutativa de les operacions per tal de no excedir el temps i fer un codi més eficient

Lletres

Per aquesta part del joc també s'ha arribat a la solució mitjançant el backtracking, per a la funció de poda s'ha realitzat el següent: amb el mètode "int quantitat(int[])", que retorna la suma de les freqüències de l'array de int (paraula en forma de freqüència de lletres), es comprova que aquesta suma sigui major o igual que la llargària de la paraula elegida per l'usuari. D'aquesta manera, només es tindran en compte les paraules que siguin una solució millor a la de l'usuari (paraules més llargues usant les lletres disponibles). Per altra banda, abans d'afegir una combinació de lletres al conjunt de paraules trobades, es comprova que aparegui al diccionari amb el mètode "Paraula apareixFreq(int[])" que fa l'esmentat retornant la paraula corresponent en cas afirmatiu o null en cas que no aparegui.